

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



TRABAJO FIN DE MÁSTER

Estudio avanzado de herramientas comerciales en HIL

Máster Universitario en Ingeniería de Telecomunicación

Autor: Pablo Amor Peinado

Tutor: María Sofía Martínez García

Ponente: Ángel de Castro Martín

Julio 2020

ESTUDIO AVANZADO DE HERRAMIENTAS COMERCIALES EN HIL

AUTOR: Pablo Amor Peinado
TUTOR: María Sofía Martínez García
PONENTE: Ángel de Castro Martín



Hardware and Control Technology Laboratory (HCTLab)
Dpto. de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio de 2020



Agradecimientos

Con este trabajo culmina otra etapa de mi trayectoria estudiantil: el máster. Se trata de una etapa bastante especial al ser, muy probablemente, el fin de mi vida como estudiante. Mentiría si dijera que este máster ha sido un camino de rosas, libre de dificultades o momentos duros, pero no nos engañemos, si echo la vista hacia atrás voy a echar mucho de menos la vida académica, y no por masoquismo, no, sino porque hay algo bastante satisfactorio en todo el proceso didáctico y, aparte, ¡es que se me da muy bien estudiar!

Me gustaría agradecer en primer lugar a los miembros del HCTLab por acogerme entre ellos para la realización tanto de mi TFG como de este TFM y por despertar mi interés y formarme en el campo de la electrónica. En especial a mi tutora Sofía, con la que siempre me he sentido muy cómodo durante todo el desarrollo de este trabajo y a Ángel, por siempre acordarse de mí. ¡Qué importante es que los profesores sean capaces de transmitir su pasión a los alumnos!

Por supuesto, no me puedo olvidar de mis padres, que siempre me han estado apoyando desde el principio y, a buen seguro, les llenará de orgullo y satisfacción el que haya concluido estos estudios del máster. Y, cómo no, agradecer mucho a mi hermano Álvaro todo el apoyo que me da todos los días.

Por último, dedicar este trabajo al grandísimo Nicolas Cage, el mejor actor del mundo.

Vaya año más surrealista para terminar, esperemos que los años siguientes sean algo más normales...

Pablo Amor Peinado

Julio 2020

Resumen

Hardware-in-the-Loop (HIL) se trata en un método efectivo y económico de simulación de circuitos digitales, en el cual se emula el funcionamiento de una parte de un sistema *hardware* en tiempo real (RT) mediante *software*, para su integración con el resto de los componentes *hardware* reales de modo que se pueda probar el comportamiento esperado del sistema físico completo.

Este Trabajo de Fin de Máster tiene como objetivo principal el estudio de las diferentes soluciones comerciales, disponibles actualmente en el mercado, que permiten llevar a cabo simulaciones HIL.

Para ello, primero se han analizado las herramientas ofertadas por cada empresa, que trabaja en este sector, hasta seleccionar aquellas que permitan realizar dichas simulaciones mediante un *software* gratuito o con versión de prueba. De cada una de estas plataformas se han analizado sus características limitantes tanto en *software* como en capacidad de *hardware*.

En un primer análisis se estudiaron las plataformas de: Typhoon HIL, LabVIEW FPGA, Simulink Real Time, Opal-RT, dSpace y Speedgoat, de las cuales se han acabado concretando las dos herramientas principales por las que se ha optado en este estudio, dado que cumplen los requisitos en cuanto a costes y *software* establecidos, que son: Typhoon HIL y LabVIEW FPGA.

Una vez realizado un primer estudio de tipo teórico sobre dichas herramientas, se decidió llevar a cabo el modelado, montaje y simulación de un circuito convertidor conmutado reductor con y sin pérdidas, equivalente para todas las plataformas, con el objetivo de establecer una comparativa más exhaustiva sobre un ejemplo práctico concreto, y así verificar los resultados teóricos obtenidos.

Palabras clave

Hardware-In-the-Loop, convertidor *buck*, simulación a tiempo real, simulación *on-line/off-line*, frecuencia de conmutación, paso de simulación, FPGA, rizado.

Abstract

Hardware-In-the-Loop (HIL) is an effective and economic simulation method for digital circuits, in which the behaviour of a part of a hardware system is emulated in real-time (RT) via software so that it can be integrated with the rest of the real hardware components to test the performance of the complete physical system.

This Project's main objective consists in studying the different possible commercial solutions currently available in the market, that are able to run HIL simulations.

To that purpose, the first step was to analyze the tools that are available for each company involved in the HIL field, in order to select those that allow the user to run these simulations on a free or trial-version software. For each one of those platforms an analysis has been made to determine their limiting software features and hardware capacity.

In a first study the following platforms were analyzed: Typhoon HIL, LabVIEW FPGA, Simulink Real Time, Opal-RT, dSpace and Speedgoat, from which the two main platforms under study in this Project were selected, as they fulfill the established requirements in terms of cost and software, and they are: Typhoon HIL and LabVIEW FPGA.

After that first theoretical study about these tools, it was decided to design, assemble and simulate a buck converter, with and without electrical losses, equivalent for both platforms so that a more exhaustive comparison can be made on a particular practical model, and thus verify the theoretical results.

Keywords

Hardware-In-the-Loop, buck converter, real time simulation, on-line/off-line simulation, switching frequency, simulation step, FPGA, ripple.

ÍNDICE DE CONTENIDOS

1 INTRODUCCIÓN.....	9
1.1 MOTIVACIÓN	9
1.2 OBJETIVOS.....	9
1.3 ORGANIZACIÓN DE LA MEMORIA	10
2 ESTADO DEL ARTE	11
2.1 TIPOS DE SIMULACIÓN.....	11
2.1.1 Simulación en tiempo real.....	12
2.1.2 Simulación on-line/off-line	12
2.1.3 Simulación Hardware-In-the-Loop (HIL).....	13
2.2 HERRAMIENTAS GENÉRICAS.....	13
2.3 EJEMPLO DE APLICACIÓN: CONVERTIDOR REDUCTOR	14
3 ESTUDIO DE HERRAMIENTAS COMERCIALES	21
3.1 TYPHOON HIL	23
3.1.1 HIL402	27
3.1.2 Typhoon HIL Control Center	29
3.2 NATIONAL INSTRUMENTS	36
3.2.1 NI myRIO-1900.....	40
3.2.2 LabVIEW Control Design and Simulation Module & Multisim	41
3.2.3 LabVIEW FPGA Module	44
3.3 MATLAB SIMULINK.....	48
3.3.1 Simulink Real Time	48
3.3.2 Simulink Real Time Desktop	48
3.4 COMPARATIVA CARACTERÍSTICAS HERRAMIENTAS COMERCIALES.....	54
4 INTEGRACIÓN, PRUEBAS Y RESULTADOS	62
4.1 CARACTERÍSTICAS BUCK	62
4.2 DISEÑO, SIMULACIÓN Y MONTAJE CONVERTIDOR CONMUTADO BUCK (SIN PÉRDIDAS)	63
4.2.1 Modelo teórico.....	63
4.2.2 Typhoon HIL	67
4.2.3 LabVIEW Control Design and Simulation Module + Multisim	77
4.2.4 LabVIEW FPGA	82
4.3 DISEÑO, SIMULACIÓN Y MONTAJE CONVERTIDOR CONMUTADO BUCK (CON PÉRDIDAS).....	95
4.3.1 Modelo teórico.....	95
4.3.2 Typhoon HIL	97
4.3.3 LabVIEW FPGA	99
4.4 COMPARATIVA DE RESULTADOS SIMULACIÓN HIL CONVERTIDOR BUCK	100
4.4.1 Buck asíncrono sin pérdidas.....	100
4.4.2 Buck asíncrono con pérdidas.....	105
4.4.3 LabVIEW simulaciones a mínimo “dt”	112
5 CONCLUSIONES Y TRABAJO FUTURO.....	115
5.1 CONCLUSIONES.....	115
5.2 TRABAJO FUTURO	117
REFERENCIAS	118
GLOSARIO	121

ÍNDICE DE FIGURAS

FIGURA 2-1: ESQUEMA DE TIPOS DE SIMULACIÓN	11
FIGURA 2-2: DISTRIBUCIÓN DEL MERCADO HIL POR PRODUCTO [4]	13
FIGURA 2-3: DISTRIBUCIÓN DEL MERCADO HIL POR FABRICANTE [5]	14
FIGURA 2-4: ESQUEMA BÁSICO FUENTE DE ALIMENTACIÓN [1]	15
FIGURA 2-5: CIRCUITO <i>BUCK</i> ASÍNCRONO (SIN PÉRDIDAS)	15
FIGURA 2-6: CIRCUITO <i>BUCK</i> ESCENARIO: <i>SWITCH</i> ON Y DIODO ABIERTO (SIN PÉRDIDAS)	15
FIGURA 2-7: CIRCUITO <i>BUCK</i> ESCENARIO: <i>SWITCH</i> OFF Y DIODO CONDUCE (SIN PÉRDIDAS)	16
FIGURA 2-8: CIRCUITO <i>BUCK</i> ESCENARIO: <i>SWITCH</i> OFF Y DIODO ABIERTO (SIN PÉRDIDAS)	16
FIGURA 2-9: CIRCUITO <i>BUCK</i> ASÍNCRONO (CON PÉRDIDAS)	17
FIGURA 2-10: CIRCUITO <i>BUCK</i> ESCENARIO: <i>SWITCH</i> ON Y DIODO ABIERTO (CON PÉRDIDAS)	17
FIGURA 2-11: CIRCUITO <i>BUCK</i> ESCENARIO: <i>SWITCH</i> OFF Y DIODO CONDUCE (CON PÉRDIDAS)	18
FIGURA 2-12: CIRCUITO <i>BUCK</i> ESCENARIO: <i>SWITCH</i> OFF Y DIODO ABIERTO (CON PÉRDIDAS)	18
FIGURA 3-1: MAPA DE DISTRIBUIDORES GLOBALES TYPHOON HIL [8]	23
FIGURA 3-2: LOGOTIPO TYPHOON HIL [7]	23
FIGURA 3-3: PANEL DE OPCIONES DE APRENDIZAJE HIL ACADEMY [9]	24
FIGURA 3-4: EJEMPLO DE CURSO HIL ACADEMY [9]	24
FIGURA 3-5: EJEMPLO DE TUTORIAL HIL ACADEMY [9]	25
FIGURA 3-6: LICENCIA CON ACCESO A MÁQUINA VIRTUAL HIL402 Y <i>TOOLBOXES</i>	26
FIGURA 3-7: <i>HARDWARE</i> HIL402 Y HIL404 [10][11]	26
FIGURA 3-8: PASOS DE SIMULACIÓN DISPONIBLES HIL402	28
FIGURA 3-9: ESPECIFICACIONES TÉCNICAS HIL402 [11]	28
FIGURA 3-10: MENÚ TYPHOON HIL CONTROL CENTER	29
FIGURA 3-11: ESQUEMÁTICO REALIZADO CON SCHEMATIC EDITOR TYPHOON HIL CONTROL CENTER	30
FIGURA 3-12: SIMULACIÓN HIL CON HIL SCADA TYPHOON HIL CONTROL CENTER	30
FIGURA 3-13: <i>SCRIPT</i> EN TYPHOON TEST IDE TYPHOON HIL CONTROL CENTER	31
FIGURA 3-14: BANCO DE PRUEBAS TEST SUITE TYPHOON HIL CONTROL CENTER	32
FIGURA 3-15: TIPOS DE <i>SOLVERS</i> DE TYPHOON HIL402	32
FIGURA 3-16: MODOS DE CONFIGURACIÓN HIL402	33
FIGURA 3-17: INCORPORACIÓN DE PÉRDIDAS AL CIRCUITO SCHEMATIC EDITOR TYPHOON HIL	33
FIGURA 3-18: REPORTE UTILIZACIÓN TYPHOON HIL	34
FIGURA 3-19: ADQUISICIÓN DE DATOS Y FORMATOS TYPHOON HIL	34
FIGURA 3-20: LOGOTIPO NATIONAL INSTRUMENTS [12]	36
FIGURA 3-21: LOGOTIPO LABVIEW [12]	36
FIGURA 3-22: EXPLORADOR DE PROYECTOS LABVIEW	37
FIGURA 3-23: EJEMPLO PANEL FRONTAL LABVIEW	38
FIGURA 3-24: EJEMPLO DIAGRAMA DE BLOQUES LABVIEW	39
FIGURA 3-25: DISPOSITIVO MYRIO-1900 CON FPGA XILINX ZYNQ-7000 [14]	40
FIGURA 3-26: LOGOTIPO MULTISIM [15]	41
FIGURA 3-27: TIPOS DE <i>SOLVERS</i> DE LABVIEW CONTROL & SIMULATION LOOP	42
FIGURA 3-28: EJEMPLO PÉRDIDAS EN CONDENSADOR MULTISIM	43
FIGURA 3-29: VISTA DE PROYECTO LABVIEW FPGA	44
FIGURA 3-30: BLOQUES ESPECÍFICOS MYRIO	45
FIGURA 3-31: EJEMPLO DE SALIDA ANALÓGICA CONECTOR C MYRIO	45
FIGURA 3-32: EJEMPLO REPORTE UTILIZACIÓN LABVIEW FPGA	46
FIGURA 3-33: LOGOTIPO MATHWORKS [17]	48
FIGURA 3-34: BLOQUE REAL-TIME SYNCHRONIZATION SIMULINK REAL TIME DESKTOP	49
FIGURA 3-35: CONFIGURACIÓN BLOQUE REAL-TIME SYNCHRONIZATION SIMULINK REAL TIME DESKTOP	49
FIGURA 3-36: ESQUEMA <i>BUCK</i> SIMULINK REAL TIME DESKTOP	50
FIGURA 3-37: MUESTREO PEQUEÑO BLOQUE REAL-TIME SYNCHRONIZATION	50
FIGURA 3-38: ERROR <i>TICKS</i> MÁXIMOS EXCEDIDOS	51
FIGURA 3-39: <i>TICKS</i> OBTENIDOS SIMULACIÓN <i>BUCK</i> REAL TIME (<i>TIME STEP</i> = 0,1 ms)	51
FIGURA 3-40: EVOLUCIÓN <i>TICKS</i> CON EL TIEMPO	51
FIGURA 3-41: TIPOS DE <i>SOLVERS</i> DE SIMULINK	52
FIGURA 3-42: EJEMPLO PÉRDIDAS EN MOSFET SIMULINK	53

FIGURA 4-1: CIRCUITO <i>BUCK</i> ASÍNCRONO CON PÉRDIDAS	62
FIGURA 4-2: SIMULACIÓN <i>BUCK</i> TEÓRICO SIMULINK (SIN PÉRDIDAS).....	63
FIGURA 4-3: GENERACIÓN PWM <i>BUCK</i> SIMULINK.....	64
FIGURA 4-4: CONFIGURACIÓN DE <i>SOLVER</i> POWERGUI SIMULINK	64
FIGURA 4-5: PARÁMETROS SIMULACIÓN IDEAL SIMULINK.....	65
FIGURA 4-6: MEDIDA V_o TEÓRICA SIMULINK (SIN PÉRDIDAS).....	65
FIGURA 4-7: MEDIDA I_L TEÓRICA SIMULINK (SIN PÉRDIDAS)	66
FIGURA 4-8: BIBLIOTECAS DE COMPONENTES Y BLOQUES TYPHOON HIL	67
FIGURA 4-9: CIRCUITO <i>BUCK</i> TYPHOON HIL (SIN PÉRDIDAS)	68
FIGURA 4-10: OBTENCIÓN <i>Duty</i> 0,41667 CON VARIAS REFERENCIAS TYPHOON HIL.....	69
FIGURA 4-11: CONFIGURACIÓN BLOQUE <i>BUCK</i> TYPHOON HIL.....	69
FIGURA 4-12: PARÁMETROS ESQUEMÁTICO TYPHOON HIL	70
FIGURA 4-13: WIDGETS HIL SCADA SIMULACIÓN <i>BUCK</i>	71
FIGURA 4-14: INICIALIZACIÓN PARÁMETROS DE ENTRADA HIL SCADA.....	72
FIGURA 4-15: SIMULACIÓN SCOPE <i>BUCK</i> HIL SCADA.....	73
FIGURA 4-16: SIMULACIÓN CAPTURE <i>BUCK</i> HIL SCADA (SIN PÉRDIDAS)	73
FIGURA 4-17: CURSOR SIMULACIÓN CAPTURE <i>BUCK</i> HIL SCADA (SIN PÉRDIDAS)	74
FIGURA 4-18: ESQUEMÁTICO 4 <i>BUCKS</i> TYPHOON HIL	75
FIGURA 4-19: UTILIZACIÓN 4 <i>BUCKS</i> TYPHOON HIL	75
FIGURA 4-20: 4 CICLOS DE TRABAJO <i>BUCKS</i>	76
FIGURA 4-21: SIMULACIÓN CAPTURE 4 <i>BUCKS</i> TYPHOON HIL.....	76
FIGURA 4-22: 4 CURSORES SIMULACIÓN CAPTURE <i>BUCK</i> TYPHOON HIL	76
FIGURA 4-23: SIMULACIÓN CIRCUITO <i>BUCK</i> MULTISIM	77
FIGURA 4-24: GENERACIÓN DE SEÑAL PWM CIRCUITO <i>BUCK</i> MULTISIM	77
FIGURA 4-25: CIRCUITO MULTISIM A EXPORTAR A LABVIEW	78
FIGURA 4-26: CIRCUITO <i>BUCK</i> LABVIEW CONTROL DESIGN AND SIMULATION + MULTISIM	79
FIGURA 4-27: GENERACIÓN DE SEÑAL PWM CIRCUITO <i>BUCK</i> LABVIEW.....	79
FIGURA 4-28: PARÁMETROS BUCLE CONTROL & SIMULATION	80
FIGURA 4-29: RESULTADO VOLTAJE DE SALIDA SIMULACIÓN CON MULTISIM	80
FIGURA 4-30: DIAGRAMA DE BLOQUES SIMULACIÓN ECUACIONES DIFERENCIALES <i>BUCK</i> EN LABVIEW	85
FIGURA 4-31: PANEL FRONTAL SIMULACIÓN ECUACIONES DIFERENCIALES <i>BUCK</i> EN LABVIEW	86
FIGURA 4-32: ERROR <i>TIMING</i> COMPONENTES BAJO NIVEL LABVIEW FPGA	87
FIGURA 4-33: EJEMPLO FEEDBACK NODE Y BLOQUE SUMADOR HIGH-THROUGHPUT DSP LABVIEW FPGA [6].....	88
FIGURA 4-34: EJEMPLO CONFIGURACIÓN HIGH-THROUGHPUT DSP LABVIEW FPGA	88
FIGURA 4-35: BLOQUE GENERACIÓN PWM LABVIEW FPGA.....	89
FIGURA 4-36: BLOQUE CÁLCULO LOOP RATE LABVIEW FPGA.....	89
FIGURA 4-37: MÓDULO SINCRONISMO LABVIEW FPGA	90
FIGURA 4-38: BLOQUE DE ESCRITURA DE SALIDA LABVIEW FPGA.....	90
FIGURA 4-39: BUCLE PRINCIPAL LABVIEW FPGA.....	92
FIGURA 4-40: PANEL FRONTAL <i>BUCK</i> LABVIEW FPGA (SIN PÉRDIDAS)	93
FIGURA 4-41: MÉTODOS DE COMPILACIÓN DE LABVIEW [23]	93
FIGURA 4-42: UTILIZACIÓN EN FASE DE SÍNTESIS Y <i>PLACEMENT</i> CIRCUITO <i>BUCK</i> LABVIEW FPGA.....	94
FIGURA 4-43: MONTAJE CIRCUITO <i>BUCK</i> LABVIEW FPGA	94
FIGURA 4-44: SIMULACIÓN <i>BUCK</i> TEÓRICO SIMULINK (CON PÉRDIDAS)	95
FIGURA 4-45: MEDIDA V_o TEÓRICA SIMULINK (CON PÉRDIDAS)	96
FIGURA 4-46: MEDIDA I_L TEÓRICA SIMULINK (CON PÉRDIDAS)	96
FIGURA 4-47: CIRCUITO <i>BUCK</i> TYPHOON HIL (CON PÉRDIDAS).....	97
FIGURA 4-48: SIMULACIÓN CAPTURE <i>BUCK</i> HIL SCADA (CON PÉRDIDAS)	98
FIGURA 4-49: CURSOR SIMULACIÓN CAPTURE <i>BUCK</i> HIL SCADA (CON PÉRDIDAS)	98
FIGURA 4-50: PANEL FRONTAL <i>BUCK</i> LABVIEW FPGA (CON PÉRDIDAS).....	99
FIGURA 4-51: COMPARATIVA V_o TYPHOON VS LABVIEW (SIN PÉRDIDAS)	100
FIGURA 4-52: DETALLE COMPARATIVA V_o TYPHOON VS LABVIEW (SIN PÉRDIDAS)	101
FIGURA 4-53: COMPARATIVA V_o TYPHOON VS LABVIEW VS SIMULINK (SIN PÉRDIDAS).....	101
FIGURA 4-54: COMPARATIVA RIZADOS V_o TYPHOON VS LABVIEW VS SIMULINK (SIN PÉRDIDAS)	102
FIGURA 4-55: COMPARATIVA I_L TYPHOON VS LABVIEW (SIN PÉRDIDAS)	103
FIGURA 4-56: DETALLE COMPARATIVA I_L TYPHOON VS LABVIEW (SIN PÉRDIDAS)	103
FIGURA 4-57: COMPARATIVA I_L TYPHOON VS LABVIEW VS SIMULINK (SIN PÉRDIDAS)	104

FIGURA 4-58: COMPARATIVA RIZADOS I_L TYPHOON VS LABVIEW VS SIMULINK (SIN PÉRDIDAS)	104
FIGURA 4-59: COMPARATIVA V_O TYPHOON VS LABVIEW (CON PÉRDIDAS).....	105
FIGURA 4-60: DETALLE COMPARATIVA V_O TYPHOON VS LABVIEW (CON PÉRDIDAS).....	106
FIGURA 4-61: COMPARATIVA V_O TYPHOON VS LABVIEW VS SIMULINK (CON PÉRDIDAS)	106
FIGURA 4-62: COMPARATIVA RIZADOS V_O TYPHOON VS LABVIEW VS SIMULINK (CON PÉRDIDAS).....	107
FIGURA 4-63: COMPARATIVA I_L TYPHOON VS LABVIEW (CON PÉRDIDAS).....	108
FIGURA 4-64: DETALLE COMPARATIVA I_L TYPHOON VS LABVIEW (CON PÉRDIDAS).....	108
FIGURA 4-65: COMPARATIVA I_L TYPHOON VS LABVIEW VS SIMULINK (CON PÉRDIDAS).....	109
FIGURA 4-66: COMPARATIVA RIZADOS I_L TYPHOON VS LABVIEW VS SIMULINK (CON PÉRDIDAS).....	109
FIGURA 4-67: V_O LABVIEW MÍNIMO “DT” SIN Y CON PÉRDIDAS (IZDA. Y DCHA. RESPECTIVAMENTE).....	112
FIGURA 4-68: RIZADO V_O LABVIEW MÍNIMO “DT” SIN Y CON PÉRDIDAS (IZDA. Y DCHA. RESPECTIVAMENTE).....	112
FIGURA 4-69: I_L LABVIEW MÍNIMO “DT” SIN Y CON PÉRDIDAS (IZDA. Y DCHA. RESPECTIVAMENTE).....	113
FIGURA 4-70: RIZADO I_L LABVIEW MÍNIMO “DT” SIN Y CON PÉRDIDAS (IZDA. Y DCHA. RESPECTIVAMENTE).....	113

ÍNDICE DE TABLAS

TABLA 3-1: TABLA COMPARATIVA <i>SOLVERS</i>	54
TABLA 3-2: TABLA COMPARATIVA ADQUISICIÓN DE DATOS	55
TABLA 3-3: TABLA COMPARATIVA FRECUENCIA DE RELOJ.....	56
TABLA 3-4: TABLA COMPARATIVA FRECUENCIA DE CONMUTACIÓN.....	56
TABLA 3-5: TABLA COMPARATIVA PASO MÍNIMO SIMULACIÓN	57
TABLA 3-6: TABLA COMPARATIVA SIMULACIÓN A TIEMPO REAL	57
TABLA 3-7: TABLA COMPARATIVA MODELADO DE PÉRDIDAS	58
TABLA 3-8: TABLA COMPARATIVA UTILIZACIÓN DE RECURSOS.....	58
TABLA 3-9: TABLA COMPARATIVA DE LICENCIAS	59
TABLA 3-10: TABLA COMPARATIVA ONLINE/OFFLINE.....	59
TABLA 3-11: TABLA COMPARATIVA USABILIDAD/DOCUMENTACIÓN	60
TABLA 4-1: UTILIZACIÓN HW ESPECÍFICO TYPHOON HIL	70
TABLA 4-2: UTILIZACIÓN <i>COREO</i> TYPHOON HIL.....	71
TABLA 4-3: UTILIZACIÓN <i>SOFTWARE</i> TYPHOON HIL	71
TABLA 4-4: ECUACIONES EN DIFERENCIAS MODELO <i>BUCK</i> ASÍNCRONO.....	82
TABLA 4-5: SOLUCIONES ECUACIONES EN DIFERENCIAS MÉTODO EULER MODELO <i>BUCK</i> ASÍNCRONO	83
TABLA 4-6: PARÁMETROS PANEL FRONTAL <i>BUCK</i> LABVIEW FPGA (SIN PÉRDIDAS)	91
TABLA 4-7: PARÁMETROS PANEL FRONTAL <i>BUCK</i> LABVIEW FPGA (CON PÉRDIDAS)	99
TABLA 4-8: COMPARATIVA ERROR MEDIO V_O E I_L	110
TABLA 4-9: COMPARATIVA ERROR MEDIO DE RIZADOS V_O E I_L	110
TABLA 4-10: COMPARATIVA DE RIZADOS V_O	111
TABLA 4-11: COMPARATIVA DE RIZADOS I_L	111
TABLA 4-12: COMPARATIVA DE RIZADOS V_O E I_L A DIFERENTES “DT” EN LABVIEW FPGA.....	114

1 Introducción

1.1 Motivación

Este Trabajo de Fin de Máster se centra en el análisis y estudio de las herramientas comerciales *Hardware-In-the-Loop* (HIL) ofertadas a día de hoy por las diferentes empresas punteras en el sector.

Hardware-in-the-Loop es una técnica ampliamente utilizada en el campo de la electrónica para probar diferentes convertidores de potencia u otros modelos en tiempo real, en la que se emula una parte del sistema físico mediante *software* y está diseñado para interactuar con el resto del sistema *hardware* real para ponerlo a prueba, de manera previa al paso a producción. De este modo se testea de manera sencilla y poco costosa toda la funcionalidad del sistema.

Actualmente en el mercado existe una gran variedad de herramientas comerciales con la capacidad de desplegar un sistema HIL. En este trabajo se llevará a cabo la comparativa de algunas de estas herramientas de diversos fabricantes, seleccionando aquellas que ofrezcan la posibilidad de realizar simulaciones HIL y muestren los resultados en tiempo real sobre un *hardware* determinado sin la necesidad de adquirir el mismo, y así evitar sus elevados costes, que pueden llegar sin problemas a las decenas de miles de euros.

Otro coste que se evita de este modo consiste en los posibles gastos de evaluación de varias herramientas que habría que asumir, donde se invertirían miles de euros para probar varias herramientas, únicamente para decantarse por una de ellas, que será la más adecuada en cada caso. Por tanto, el interés del trabajo no solo estaría dedicado a estudiantes que deseen iniciarse en las simulaciones HIL, sino a todo el ámbito académico, la investigación e incluso empresas que estén pensando adquirir alguna herramienta concreta.

De las herramientas escogidas, se expondrán las diferencias en cuanto a su *software* de diseño y a las capacidades *hardware* de cada una de ellas, usando estudios similares como soporte. También se expresará una opinión subjetiva en cuanto a la usabilidad de cada una de ellas tras su empleo y familiarización.

Finalmente se simulará un convertidor conmutado reductor o *buck* partiendo de las bases establecidas en mi propio Trabajo de Fin de Grado [1], para analizar estas diferencias sobre un circuito específico, equivalente para todas las plataformas. La elección de un convertidor conmutado como circuito a simular es, además, de especial interés dado su uso frecuente y extendido como fuentes de alimentación en numerosos sistemas electrónicos.

1.2 Objetivos

Los objetivos definidos en términos generales durante el inicio de este Trabajo de Fin de Máster han permanecido presentes durante todo su transcurso, pero, además, han ido apareciendo nuevos retos, ligados en su mayoría al producto ofertado por cada empresa y a

su disponibilidad, que han añadido nuevos objetivos. Por ello, el listado definitivo de los mismos queda finalmente de la siguiente forma:

- 1.) Estudio y familiarización con la terminología relacionada con *Real Time* (RT) y HIL.
- 2.) Análisis de las empresas involucradas en el sector de dichas tecnologías.
- 3.) Estudio de las herramientas ofertadas por cada empresa para el diseño y simulación de tipo HIL de un modelo determinado.
- 4.) Comparativa de las capacidades y limitaciones de cada una de las plataformas de manera teórica a través de sus características y su análisis a nivel de usuario.
- 5.) Observación y verificación de los resultados teóricos mediante la simulación de un convertidor conmutado concreto equivalente en todas las plataformas, sin pérdidas y con pérdidas.

1.3 Organización de la memoria

La memoria de este trabajo se encuentra estructurada de la siguiente forma:

- En el capítulo 2 se ilustran las características principales que debe tener una simulación para ser en tiempo real, *on-line/off-line* y *Hardware-In-the-Loop*, de cara a establecer la terminología empleada durante el resto del trabajo. Se comentan también en este punto las principales herramientas genéricas que se encuentran en el campo de las simulaciones HIL a día de hoy y, además, se explica el motivo por el cual se ha escogido un convertidor conmutado de tipo reductor para realizar las pruebas en cada plataforma.
- En el capítulo 3 se comienza detallando las limitaciones encontradas a la hora de seleccionar una herramienta HIL para probar y, a continuación, se procede a analizar y comparar cada una de las herramientas escogidas, en términos de *software* y capacidad *hardware* según sus especificaciones.
- En el capítulo 4 se lleva a cabo el modelado y simulación HIL de un mismo convertidor *buck* en cada una de las plataformas, con y sin pérdidas, para estudiar sus características, limitaciones y comportamiento sobre un caso específico.
- Por último, en el capítulo 5 se termina analizando las conclusiones extraídas durante el trabajo y se plantean las líneas futuras que deja abiertas tras su finalización a un posterior análisis o investigación.

2 Estado del arte

2.1 Tipos de simulación

Cuando se habla de simular un circuito a lo que en realidad se está haciendo alusión es a la traducción del comportamiento del modelo circuital en un modelo matemático, normalmente a través de su descomposición en ecuaciones diferenciales ordinarias o EDOs, de cara a estudiar su funcionamiento.

Para que un *software* sea capaz de llevar a cabo una simulación, es por tanto necesario declarar un método de resolución de dichas ecuaciones, también llamado *solver*, como Euler o Runge-Kutta, y dependiendo del método escogido, los resultados de la simulación variarán en cada caso.

Además, dependiendo de su naturaleza, existen varios tipos de simulación y hay varias maneras de clasificar dichos tipos, según sus características. En la Figura 2-1 se muestra una categorización realizada en un artículo de *Energies* [2], en el que las simulaciones posibles se desgranán según si son en tiempo real, en línea y si se realizan sobre *hardware*. El camino que se abarca en este trabajo, correspondiente a las simulaciones *Hardware-In-the-Loop* se ha resaltado en verde.

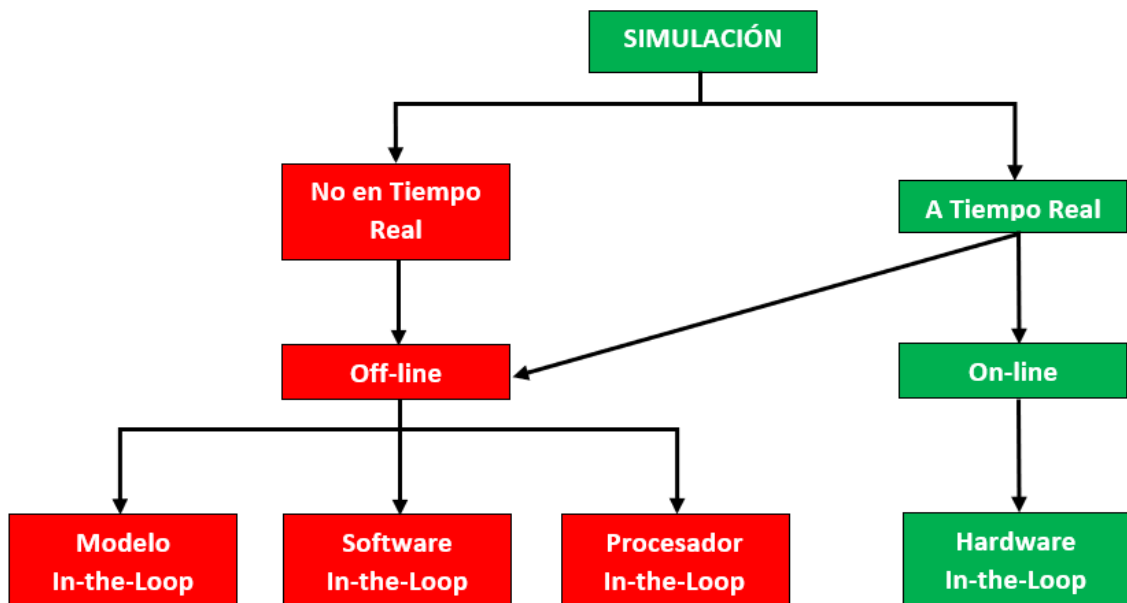


Figura 2-1: Esquema de tipos de simulación

A continuación, se detallan las particularidades de cada uno de estos tipos de simulaciones y la razón de la diferenciación en cada nivel del gráfico, profundizando especialmente en todo lo relacionado a las simulaciones HIL.

2.1.1 Simulación en tiempo real

La primera característica para diferenciar una simulación consiste en determinar si dicha simulación se va a ejecutar, o no, en tiempo real.

La diferencia fundamental entre una simulación a tiempo real y otra que no se ejecuta a tiempo real subyace en el hecho de que una simulación a tiempo real (RT), se encuentra sometida a unos tiempos determinados. Es decir, mientras que una simulación que no funciona a tiempo real no tiene limitaciones ni plazos temporales, una simulación RT tiene que ejecutarse en unos instantes determinados.

Este hecho no significa que la simulación del sistema tenga que ejecutarse necesariamente con mayor rapidez, sino que debe proporcionar el dato en el momento adecuado, de modo que simule consistentemente la evolución de la dinámica del sistema que ocurriría en tiempo no de simulación, sino externo, del mundo real.

Para lograr que esto sea así, es crítico que los sistemas de tiempo real sean capaces de trabajar de forma concurrente con varias tareas ejecutándose simultáneamente, entre las cuales puede haber dependencias de varios tipos, como que una no pueda empezar sin que otra acabe. Esto conlleva un gran trabajo de sincronización y de disponibilidad de recursos con los que debe lidiar la plataforma sobre la que se lance la simulación.

En la actualidad es posible desplegar simulaciones a tiempo real sobre un ordenador con un procesador adecuado, también sobre FPGAs u otro *hardware* similar [2].

2.1.2 Simulación *on-line/off-line*

El segundo nivel de diferenciación se trata de comprobar si la simulación es *on-line* o, por el contrario, es *off-line*.

En una simulación *on-line*, el sistema simulado pasa a formar parte del proceso completo mientras dura la simulación, mientras que en el caso de una simulación *off-line*, la simulación se ejecuta de manera independiente.

Por tanto, es evidente que una simulación *on-line* solamente pueda tener lugar si es además una simulación a tiempo real ya que como en una simulación RT se tiene que cumplir la sincronización de datos y de entradas y salidas con el tiempo del mundo real, es posible que el sistema diseñado se integre en el proceso durante la simulación, pudiendo lidiar con cambios de parámetros en cualquier momento determinado siendo, por tanto, además, una simulación *on-line*.

También existe la posibilidad de una simulación RT *off-line*, en la que, pese a cumplirse los sincronismos típicos de la simulación a tiempo real el sistema se ejecuta de manera independiente, pero no es muy frecuente su empleo, ya que, si no se va a interactuar a tiempo real con el sistema, no tiene sentido normalmente el empleo de RT [2].

2.1.3 Simulación *Hardware-In-the-Loop* (HIL)

Por último, la diferenciación final hace referencia a dónde se lleva a cabo el despliegue del sistema. En caso de llevarse a cabo una simulación de tipo *off-line*, ésta puede ser de tipo modelo, procesador o *Software-In-the-Loop*, mientras que, si se hace una simulación a tiempo real y se despliega sobre un *hardware* determinado, entonces se habla de una simulación *Hardware-In-the-Loop* (HIL).

Todo esto conlleva, que mientras que una simulación tradicional *off-line* y no RT, resuelve las ecuaciones en diferencias del modelo de manera ideal, normalmente conllevando grandes tiempos de simulación, una simulación *on-line*, a RT y HIL, no solo se debe ajustar a un paso temporal o *time step* determinado, y debe ser posible interactuar a tiempo real con las señales durante la simulación, sino que además se despliega sobre un *hardware* específico.

De este modo, es fácil comprender por qué las simulaciones HIL están siendo tan bien recibidas, ya que permiten observar los resultados que se obtendría al integrar un sistema con el *hardware* deseado y probar todo tipo de escenarios, en un entorno seguro y de bajo coste. Como ejemplo, en la industria automovilística, empresas como Mazda [3], ya han optado por el empleo de estas tecnologías para testear y depurar el diseño de los circuitos electrónicos de sus coches, de modo que se puedan realizar tests de todo tipo en situaciones difícilmente simulables de otro modo, como altas velocidades, curvas o pendientes. Así se puede interactuar a tiempo real con el controlador central del automóvil, todo esto previo al paso de producción de dichos circuitos, que se lleva a cabo una vez todo está perfectamente probado, lo cual es altamente eficiente y abarata enormemente los costes [2].

2.2 Herramientas genéricas

Actualmente existe un buen número de empresas que, o bien ofrecen ya sus propias herramientas y máquinas dedicadas a las simulaciones HIL, o aplican otras comunes en sus diferentes proyectos. Como se puede ver en el estudio del mercado de este sector realizado en enero de 2020 [4] e ilustrado en la Figura 2-2, es el sector automovilístico el que más partido saca las tecnologías HIL, seguido de cerca por investigación y educación, la electrónica de potencia y el sector aeroespacial. Además, se puede observar que se trata de un mercado en expansión, con vistas de duplicar su presencia de aquí a cinco años.

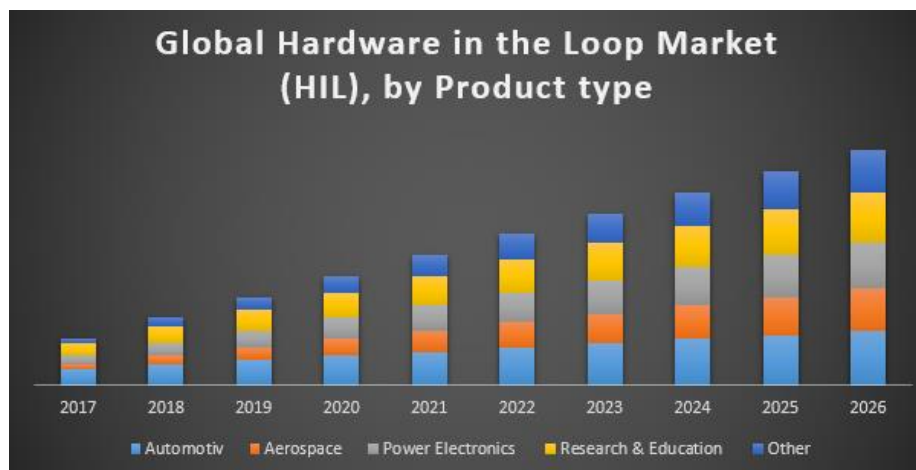


Figura 2-2: Distribución del mercado HIL por producto [4]

En cuanto a compañías específicas que trabajan en este sector, las principales aparecen representadas en la Figura 2-3 [5] según su posición en el mercado, y son fundamentalmente las siguientes:

- dSpace
- National Instruments
- Typhoon HIL
- Opal-RT
- Speedgoat
- RTDS Technologies
- Aegis Technologies Group
- Siemens
- Airbus
- Honda Aircraft Technologies
- Robert Bosch

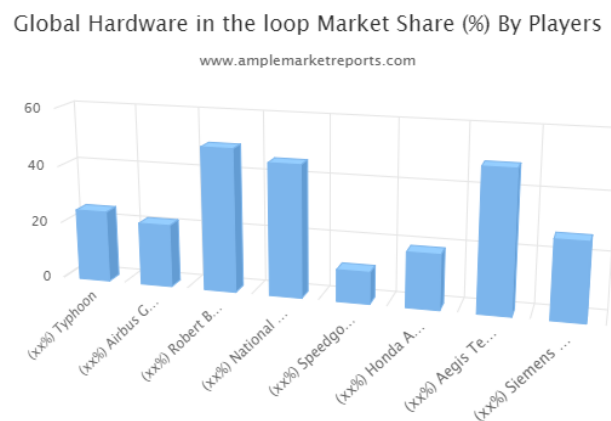


Figura 2-3: Distribución del mercado HIL por fabricante [5]

2.3 Ejemplo de aplicación: Convertidor reductor

Ya que una de las principales aplicaciones de las simulaciones *Hardware-In-the-Loop* consiste en todo lo relacionado al área de la electrónica de potencia, a la hora de probar de manera práctica las diferentes herramientas comerciales ofertadas por las empresas ya mencionadas, se ha creído conveniente partir de un convertidor conmutado.

Este tipo de circuito, como toda fuente de alimentación, cuenta con una serie de parámetros que se deben escoger para lograr el comportamiento esperado, como el hecho de si se desea en la carga un voltaje superior o inferior al de entrada, la corriente, la eficiencia o el ruido máximo permitido. En la Figura 2-4 se ilustra el esquema más simplificado posible de una fuente de alimentación, en el que existen tres elementos básicos: una tensión de entrada, una fuente de alimentación y una carga. La fuente, a través de la tensión suministrada a la entrada, pone a la salida, en la carga, un voltaje determinado. Nótese que, si bien la fuente se ha dibujado a modo de “caja negra”, en realidad está conformada normalmente por una serie de componentes que se encargan de rectificar, regular, filtrar y regular la señal de entrada, para conseguir una salida lo más constante y lineal posible.

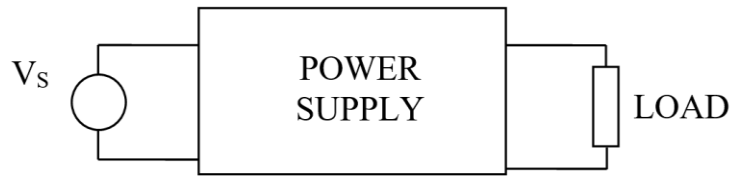


Figura 2-4: Esquema básico fuente de alimentación [1]

La opción de escoger un convertidor conmutado surge de su mayor eficiencia, que puede llegar sin problemas al 95%, respecto a un regulador lineal, aunque son algo más susceptibles al ruido ya que la salida no es corriente continua pura.

En particular, se ha empleado un convertidor conmutado reductor o *buck*, en el cual la tensión de salida es menor que la de entrada, frente al convertidor *boost*, en el que la tensión de salida es mayor que la de entrada. En la Figura 2-5 se muestra el esquema de un circuito *buck* asíncrono sin pérdidas.

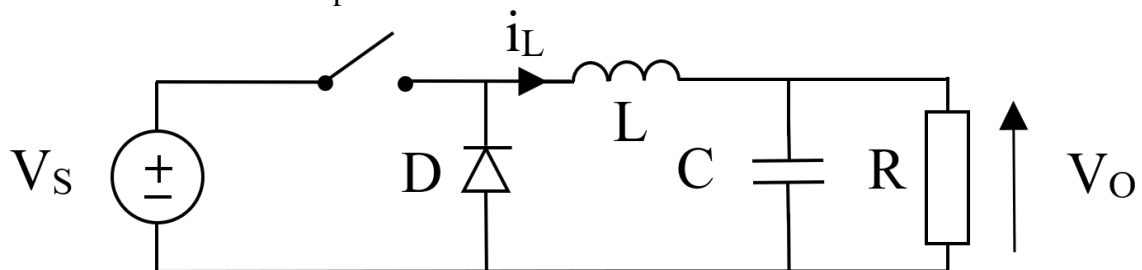


Figura 2-5: Circuito *buck* asíncrono (sin pérdidas)

El funcionamiento de este circuito se basa en el empleo de un interruptor como método de conmutación o *switching* y un diodo. Además, el ciclo de trabajo o *Duty cycle* que se emplee en cada caso, determina el valor de tensión (o corriente) de salida que se obtendrá y será necesario su ajuste para lograr en la carga el valor deseado.

Para el caso de un *buck* trabajando en modo discontinuo existen tres escenarios de funcionamiento posibles que se ilustran a continuación, en las Figuras 2-6, 2-7 y 2-8. Junto a cada circuito se muestran las ecuaciones en diferencias que modelan su comportamiento.

1.) SWITCH "ON" & DIODO "ABIERTO"

- **Circuito**

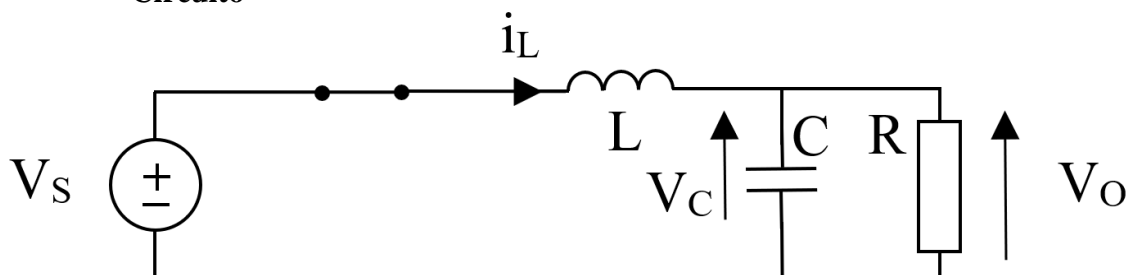


Figura 2-6: Circuito *buck* escenario: *switch* on y diodo abierto (sin pérdidas)

- Ecuaciones diferenciales

$$\frac{\partial i_L}{\partial t} = \frac{V_S - V_O}{L}$$

$$\frac{\partial V_C}{\partial t} = \frac{1}{C} \left(i_L - \frac{V_O}{R} \right)$$

$$V_O = V_C$$

2.) SWITCH “OFF” & DIODO CONDUCE

- Circuito

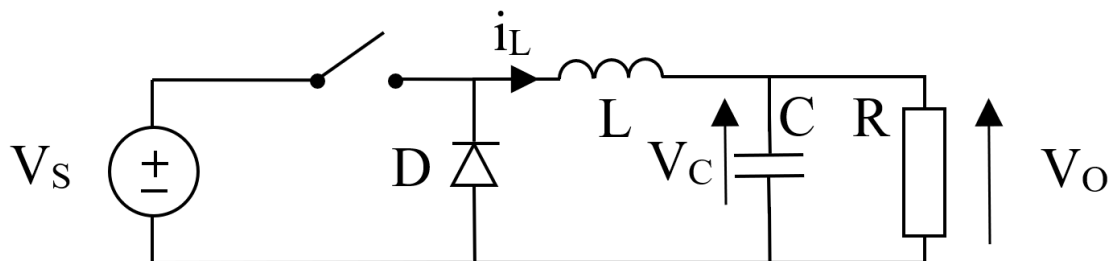


Figura 2-7: Circuito *buck* escenario: *switch* off y diodo conduce (sin pérdidas)

- Ecuaciones diferenciales

$$\frac{\partial i_L}{\partial t} = \frac{-V_O}{L}$$

$$\frac{\partial V_C}{\partial t} = \frac{1}{C} \left(i_L - \frac{V_O}{R} \right)$$

$$V_O = V_C$$

3.) SWITCH “OFF” & DIODO “ABIERTO”

- Circuito

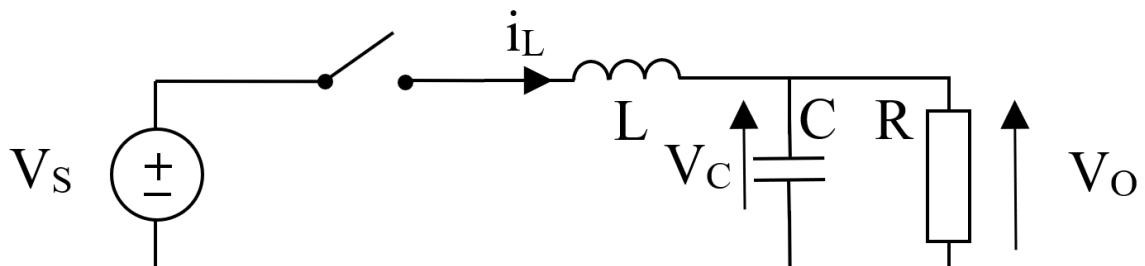


Figura 2-8: Circuito *buck* escenario: *switch* off y diodo abierto (sin pérdidas)

- **Ecuaciones diferenciales**

$$\frac{\partial i_L}{\partial t} = 0$$

$$\frac{\partial V_C}{\partial t} = -\frac{V_O}{C R}$$

$$V_O = V_C$$

Para el caso del circuito *buck* con las pérdidas que se han seleccionado para su estudio en apartados posteriores, el circuito quedaría como se ilustra en la Figura 2-9.

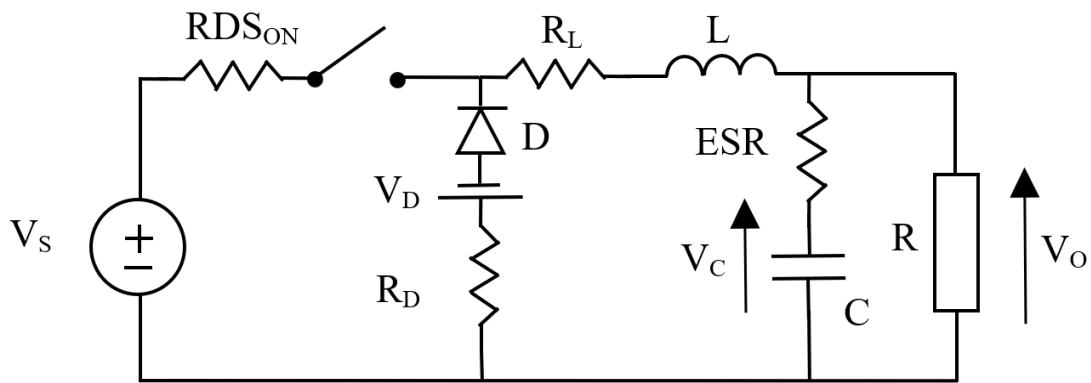


Figura 2-9: Circuito *buck* asíncrono (con pérdidas)

A continuación, se muestran nuevamente los tres escenarios posibles en modo discontinuo (Figuras 2-10, 11 y 12) junto a las ecuaciones diferenciales que lo modelan en cada caso [6].

1.) SWITCH “ON” & DIODO “ABIERTO”

- **Circuito**

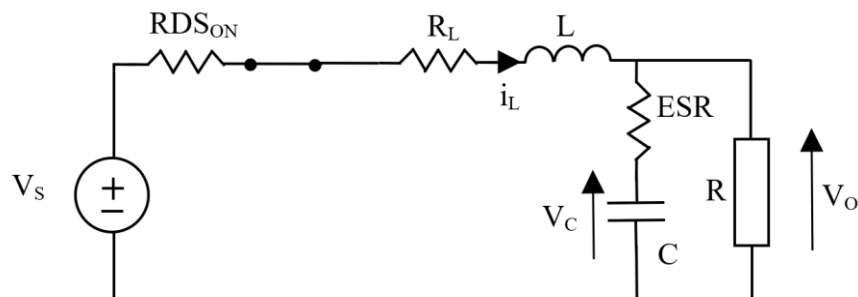


Figura 2-10: Circuito *buck* escenario: *switch* on y diodo abierto (con pérdidas)

- Ecuaciones diferenciales

$$\frac{\partial i_L}{\partial t} = \frac{V_S - i_L(R_L + R_{DS_{ON}}) - V_O}{L}$$

$$\frac{\partial V_C}{\partial t} = \frac{1}{C} \left(i_L - \frac{V_O}{R} \right)$$

$$V_O = \frac{R}{R + ESR} (i_L ESR + V_C)$$

2.) SWITCH “OFF” & DIODO CONDUCE

- Circuito

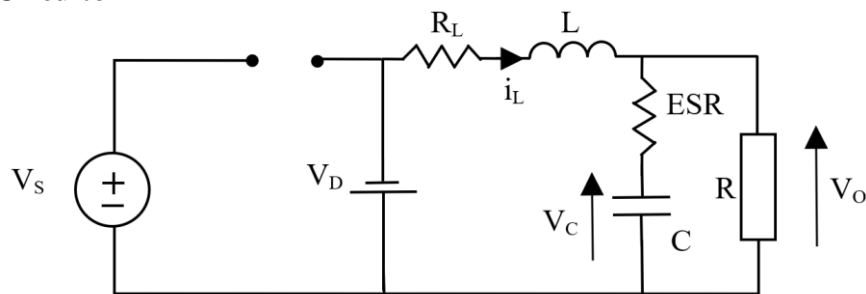


Figura 2-11: Circuito *buck* escenario: *switch* off y diodo conduce (con pérdidas)

- Ecuaciones diferenciales

$$\frac{\partial i_L}{\partial t} = \frac{-i_L R_L - V_O - V_D}{L}$$

$$\frac{\partial V_C}{\partial t} = \frac{1}{C} \left(i_L - \frac{V_O}{R} \right)$$

$$V_O = \frac{R}{R + ESR} (i_L ESR + V_C)$$

3.) SWITCH “OFF” & DIODO “ABIERTO”

- Circuito

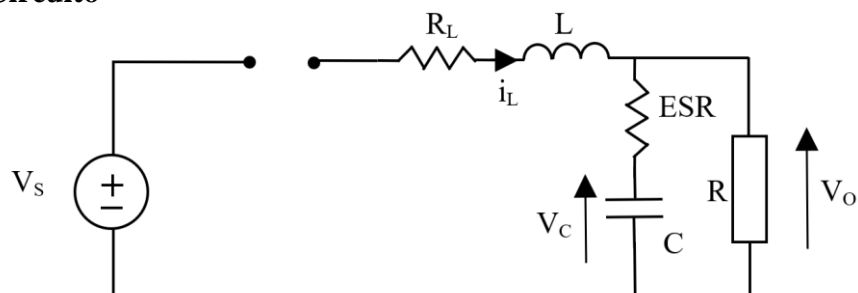


Figura 2-12: Circuito *buck* escenario: *switch* off y diodo abierto (con pérdidas)

- **Ecuaciones diferenciales**

$$\frac{\partial i_L}{\partial t} = 0$$

$$\frac{\partial V_C}{\partial t} = -\frac{V_o}{C R}$$

$$V_o = \frac{V_C R}{R + ESR}$$

3 Estudio de herramientas comerciales

En este apartado se procederá a mostrar en profundidad las diferentes opciones que se encuentran actualmente en el mercado a la hora de elegir una herramienta comercial determinada para realizar una simulación *Hardware-In-the-Loop* (HIL). Además, se analizarán y se compararán las características principales de una selección de dichas herramientas de forma tanto objetiva, mediante datos tales como: capacidad de la herramienta, rango de frecuencias soportado o tipos de licencia; como subjetiva, mediante una valoración personal sobre la usabilidad o la cantidad y calidad de la documentación disponible para cada una de ellas.

Como se mencionó con anterioridad en el apartado 2.2 las empresas o firmas más importantes que se dedican de forma activa en la actualidad a los campos de las simulaciones tanto en RT como HIL son: Typhoon HIL, Opal-RT, dSpace, National Instruments, RTDS Technologies y Speedgoat.

Dada la gran variedad de herramientas disponibles en el mercado, el querer abordar el análisis y la comparativa de todas y cada una de ellas en un único trabajo resultaría una tarea muy ambiciosa y que se escapa al alcance de este. Además, en este proyecto se contaba con una serie de limitaciones que se resumen fundamentalmente en dos:

- No se dispone de *hardware* de ninguna de estas marcas para realizar las simulaciones (salvo de National Instruments).
- El *software* que se utilizará para estudiar cada herramienta será la versión gratuita o de prueba que ofrezca cada compañía, si es que disponen de este modelo de servicio.

La decisión para llevar a cabo el estudio de esta manera no se trata de una mera cuestión de falta de recursos disponibles, sino que se ha optado por escoger un enfoque que refleje fielmente la situación que se puede encontrar un usuario medio o un estudiante que decida montar una simulación a tiempo real o en *Hardware-In-the-Loop* y no disponga de las decenas de miles de euros que puede suponer adquirir cada uno de estos *hardwares*. Además, también este estudio supone la evaluación de varias herramientas, lo que ayuda a la hora de elegir una de ellas, sin necesidad de invertir miles de euros en adquirir todas las disponibles, a ciegas, solo para elegir finalmente la más adecuada para cada usuario. Por ello, no es únicamente de interés para estudiantes o posibles usuarios que deseen iniciarse en el mundo de las simulaciones HIL, sino para todo el campo académico y de investigación, e incluso para empresas que deseen llevar a cabo un despliegue de características similares y tengan dudas sobre qué plataforma les conviene emplear.

Por tanto, se parte de que las herramientas sean capaces, en su versión de *software* gratuito, de realizar este tipo de simulaciones y, sobre todo, de mostrar los recursos, la utilización y, en definitiva, el rendimiento que supondría su despliegue en un *hardware* determinado sin necesidad de adquirirlo.

De este modo, de los fabricantes mencionados anteriormente, se descartan: Opal-RT, dSpace y RTDS Technologies. Speedgoat como tal tampoco se estudiará debido a que

requiere de su *hardware* específico para realizar las simulaciones, pero se hablará de Simulink Real Time, herramienta sobre la que Speedgoat y otro gran número de máquinas basan su empleo.

Cabe destacar que me he puesto en contacto con todas y cada una de las empresas mencionadas anteriormente, para ver si disponían de *software* de prueba o gratuito que fuera capaz de mostrar los resultados de una simulación HIL real, sin necesidad de adquirir un *hardware* concreto y, en ello, se origina una de las mayores dificultades de este trabajo y, es que, en ciertos casos no se ha obtenido respuesta de ningún tipo por parte de alguna empresa, pese a reiterados intentos, y en otros tantos me informaron de que era necesario disponer, previa compra, de un *hardware* físico. De este modo, también un factor limitante para la elección es el hecho de la disponibilidad de las plataformas.

Por tanto, las tres herramientas sobre las que se realizará el análisis serán las de Typhoon HIL, National Instruments y MATLAB Simulink, centrándose especialmente en las dos primeras.

3.1 Typhoon HIL

Typhoon HIL [7] es una de las empresas líderes en el sector de las simulaciones para controladores *Hardware-In-the-Loop* (C-HIL), sistemas de electrónica de potencia, redes de distribución y *microgrids*. La compañía surgió en 2008 y su sede principal se encuentra en Somerville, Massachusetts (EEUU) con unos 50 empleados, contando además con oficinas distribuidas por diversos países europeos y con distribuidores en tres continentes (América, Europa y Asia), siendo uno de ellos la empresa madrileña Megacal Instruments Ibérica S.L.



Figura 3-1: Mapa de distribuidores globales Typhoon HIL [8]

Desde la empresa aseguran que los diferentes productos que ofertan son de la más alta calidad y sus herramientas poseen una gran facilidad de empleo, además de ser punteras desde el punto de vista tecnológico y cuentan con un rendimiento no superado en el sector.



Figura 3-2: Logotipo Typhoon HIL [7]

Entre las aplicaciones de sus productos destaca el testeo de control e interoperabilidad en sistemas tales como convertidores o *microgrids*, y la gama de productos que tienen a la venta en el mercado incluye desde dispositivos o máquinas virtuales, hasta sistemas que trabajan a tiempo real y accesorios para HIL.

Cabe destacar también que Typhoon pone a disposición de sus usuarios una plataforma formativa de aprendizaje para todo aquel que desee adquirir conocimientos acerca del diseño de diversos modelos para la realización de simulaciones *Hardware-In-the-Loop* (HIL) y de cómo emplear sus herramientas para tales fines.

Dicha plataforma web lleva por nombre HIL Academy [9] y permite, de manera totalmente gratuita para su uso académico, apuntarse a una gran variedad de cursos sobre las herramientas de Typhoon, en los que además se otorgan una serie de certificaciones. También permite visionar tutoriales para aprender los principios básicos o descargar modelos ya diseñados para su posterior análisis o empleo a modo de base para un diseño más complejo y personalizado.

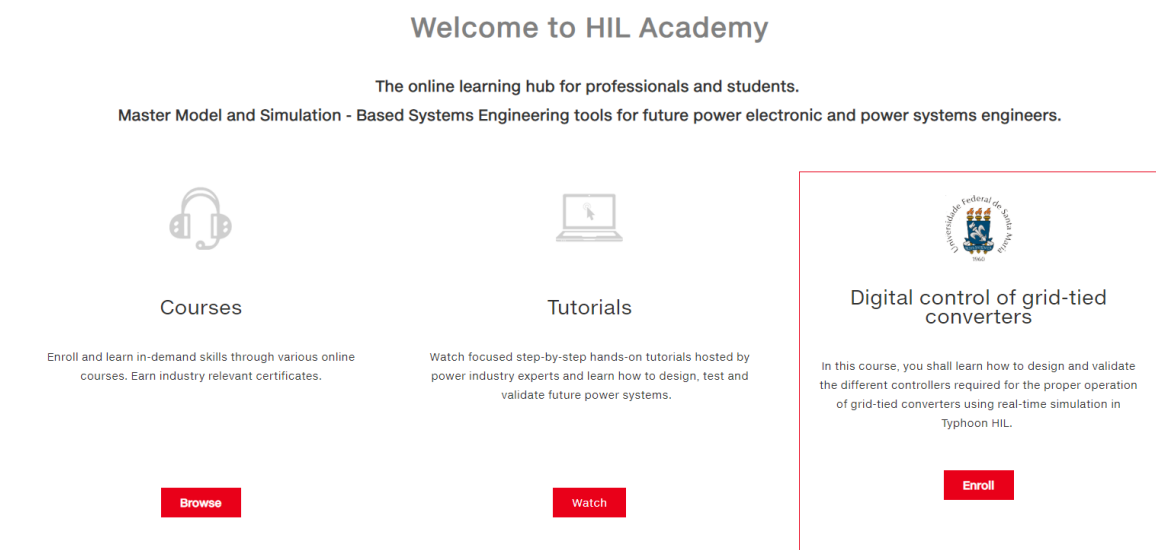


Figura 3-3: Panel de opciones de aprendizaje HIL Academy [9]

Un ejemplo de curso para la obtención de una certificación de especialista en HIL se presenta en la Figura 3-4, donde se puede observar la descripción del curso y los puntos que se estudiarán, así como las horas requeridas para su finalización y los recursos que se emplearán.

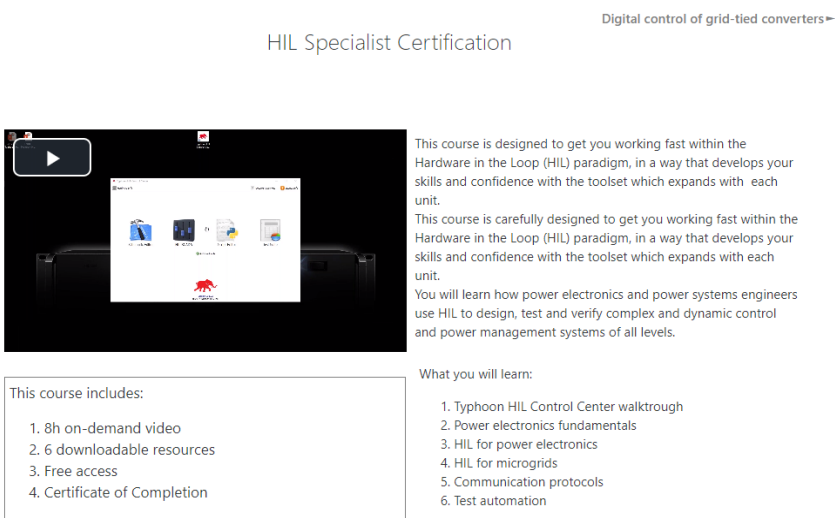


Figura 3-4: Ejemplo de curso HIL Academy [9]

En la Figura 3-5 se muestra uno de los tutoriales disponibles en la HIL Academy que se usó durante la realización de este trabajo para familiarizarse con las herramientas y que muestra desde cero el diseño, despliegue y la simulación de un sistema en lazo cerrado (en particular un control en lazo cerrado de un convertidor *boost*) a través de una serie de vídeos guiados de duraciones variables.

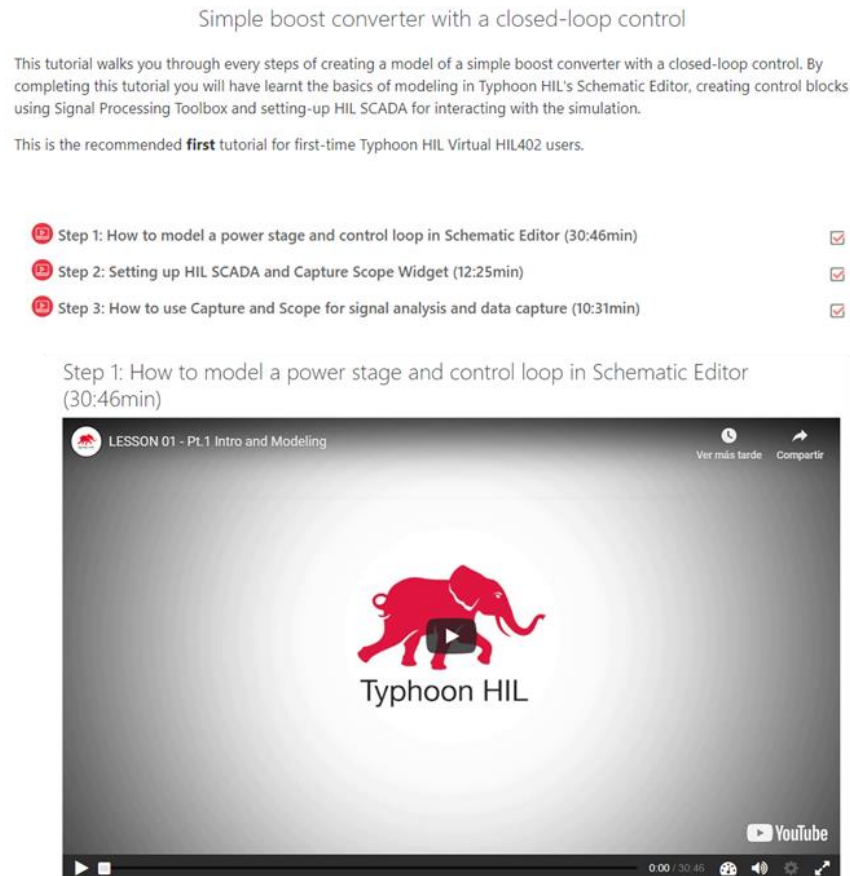


Figura 3-5: Ejemplo de tutorial HIL Academy [9]

Una vez que se realiza el registro en la página oficial de Typhoon HIL, la empresa otorga al usuario acceso a la descarga de su *software* de diseño de forma totalmente gratuita y, además, se incluye una licencia de uso de una determinada máquina virtual de un año de duración.

Dicha máquina virtual permite la simulación de los modelos diseñados en HIL de forma remota, de modo que no es necesario disponer de un *hardware* físico para simular en el lado del usuario. Por ello, únicamente es necesario llevar a cabo el modelado del sistema y, posteriormente se mandará el esquemático y el resto de los parámetros de diseño a la máquina virtual donde se realizará la simulación deseada.

En la Figura 3-6 se puede observar un ejemplo de una licencia conseguida mediante el registro en Typhoon HIL, en el que se destaca el acceso durante un año a la máquina virtual HIL402 y a las *toolboxes* de procesamiento de señales y de sistemas de potencia de forma indefinida.

Personal Data

Name:

Email:

Password:

Change password

License

Download license file

Subscriptions

Type	Start	End	Activation key	Serial no	Status
HIL402 / 1 yr	2019-03-31	2020-03-31		Virtual device	active

Toolbox Packages

Toolbox	Start	End	Status
---------	-------	-----	--------

Toolboxes

Toolbox	Start	End	Status
Signal Processing Toolbox	2016-10-31	2115-10-31	active
Power Systems Toolbox	2016-10-31	2115-10-31	active

Figura 3-6: Licencia con acceso a máquina virtual HIL402 y *toolboxes*

La máquina *hardware* a la que se tuvo acceso durante el comienzo de la elaboración de este trabajo fue la HIL402 que es sobre la que se realizará el análisis y comparativa. Sin embargo, cabe destacar que la empresa ya en 2020 ofrece acceso a su nueva máquina HIL404 [10] que mejora la anterior entre otras cosas en el *time step* (paso temporal) que es de tan solo 200 ns, frente a 0,5 μ s en el modelo anterior HIL402.

Se calcula que el nuevo modelo HIL404 es un 75% más rápido a nivel de procesador principal FPGA, diez veces más rápido a nivel del coprocesador CPU y 2,5 veces más rápido en cuanto a la entrada/salida (IO) tanto digital como analógica. Con estos datos queda demostrado lo rápido que está evolucionando todo este sector en la actualidad.

Una imagen del aspecto físico de estos sistemas se puede ver en la Figura 3-7.



Figura 3-7: Hardware HIL402 y HIL404 [10][11]

A continuación, se detallarán algunas de las características principales de la máquina que ha sido empleada (HIL402) y posteriormente se pasará a comentar el *software* propietario de diseño que ofrece la empresa para el diseño y simulación llamado Typhoon HIL Control Center.

3.1.1 HIL402

La máquina HIL402 [11] destaca por ser el primer sistema de *Hardware-In-the-Loop* (HIL) de tamaño *notebook*. Se trata por tanto de un modelo altamente compacto especialmente enfocado al testeo o simulación de controladores del campo de la electrónica de potencia como bien pueden ser generadores de potencia solares o eólicos, almacenamiento de batería o motores.

Este *hardware*, que cuenta con 4 núcleos o *cores*, es especialmente interesante en el campo de la docencia, ya que permite, junto al *software* Typhoon Control Center, el diseño gráfico del circuito, el control mediante SCADA y la simulación a tiempo real a gran velocidad.

A continuación, se presentan las características técnicas principales de HIL402, destacando aquellas en las que se ha centrado este proyecto para su comparación con otras soluciones comerciales presentes en el mercado:

- Procesador XC7Z030 Zync SoC de 4 núcleos.
- Periodo de reloj 20 ns, esto implica una frecuencia máxima de 50 MHz.
- Frecuencia de conmutación máxima PWM de 200 kHz para convertidores.
- Paso de simulación variable de 0,5 μ s a 20 μ s.
Este dato puede observarse en la Figura 3-8.
- Muy baja latencia, solo 20 ns para muestrear, calcular la nueva salida y actualizar los valores.
- Osciloscopio de 32 canales integrado.
- 32 canales digitales y 16 canales analógicos.
- Compatibilidad *software*: Typhoon Control Center y MATLAB Simulink.
- Rango de voltaje de ± 10 V.
- Conectividad mediante USB, Ethernet o máquina virtual.

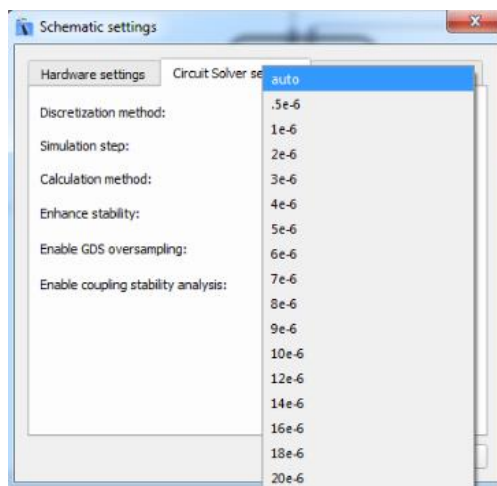


Figura 3-8: Pasos de simulación disponibles HIL402

El resto de las características y detalles técnicos se puede consultar en la Figura 3-9.

HIL402 technical details.

Processor	XC7Z030 Zync SoC	Up to 4 cores
Analog I/O	Channels	16 inputs / 16 outputs
	Resolution	16 bit
	Voltage range	± 10 V
	Sample rate	1 MSPS
	Protection	± 24 V tolerant, ESD protection
	Connector	DIN 41612, type C, 96 pin male connector
Digital I/O	Channels	32 inputs / 32 outputs
	Threshold voltages (DI)	$V_{IL(max)} = 1.5$ V; $V_{IH(min)} = 3.5$ V
	Threshold voltages (DO)	$V_{OL(max)} = 0.2$ V; $V_{OH(min)} = 4.8$ V
	Protection	± 24 V tolerant, ESD protection
	Connector	DIN 41612, type C, 96 pin male connector
Externally available power supply	± 5 V analog	up to 1 A, resettable protection
	± 12 V analog	up to 0.5 A, resettable protection
	+ 3.3 V digital	up to 1 A, resettable protection
	+ 5 V digital	up to 1 A, resettable protection
Connectivity	USB	2.0 high speed, B-type connector
Compatibility	HIL DSP interface	
	HIL uGrid DSP interface	
	HIL Breakout board	
	HIL dSpace interface board	
Power supply	External	100 - 250 VAC, ≥ 60 W

Figura 3-9: Especificaciones técnicas HIL402 [11]

3.1.2 Typhoon HIL Control Center

Typhoon HIL Control Center es el *software* que ofrece la empresa Typhoon, tras registrarse de forma gratuita, y que consiste en la integración de todos los componentes y herramientas *software* necesarios en un único programa.

En la Figura 3-10 se puede observar el menú que aparece una vez ejecutado este programa, así como los diferentes módulos cuya funcionalidad se irá desgranando a continuación.

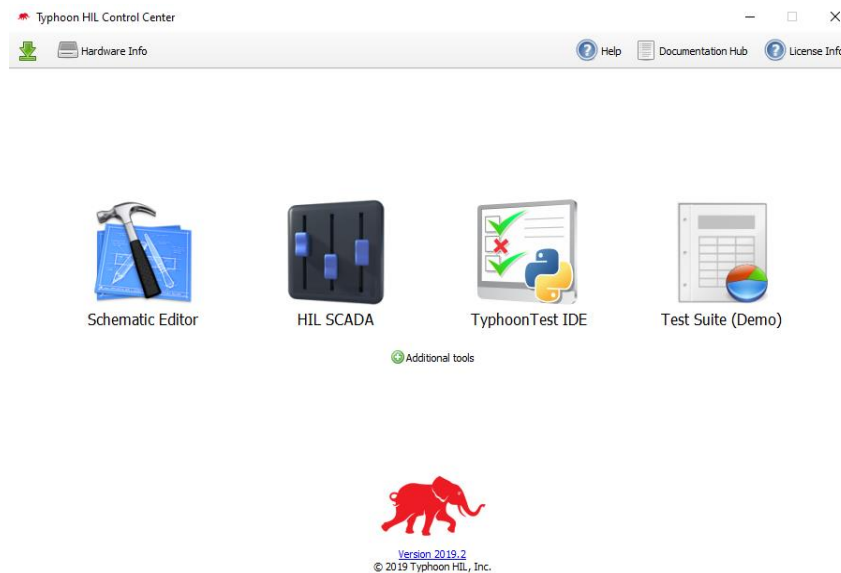


Figura 3-10: Menú Typhoon HIL Control Center

En la esquina superior derecha es posible consultar la información acerca de la licencia de la que disponemos, la documentación disponible y la opción de consulta de ayuda. En la esquina superior izquierda se puede ver si existen actualizaciones disponibles del *software* y la información de cualquier *hardware* conectado.

Los cuatro grandes módulos que abarca este programa son los siguientes:

- **Schematic Editor:**

Es la herramienta de diseño por excelencia de las que dispone Typhoon HIL Control Center. Su empleo sigue un gran parecido al de herramientas del tipo simuladores SPICE (*Simulation Program with Integrated Circuits Emphasis*) y de forma realmente cómoda y ágil permite al usuario diseñar circuitos electrónicos de alta complejidad arrastrando, depositando en el esquemático, dando valores e interconectando los diferentes componentes y bloques disponibles. Cabe destacar que ofrece una gran variedad de dichos componentes, entre los que destacan bloques de electrónica de potencia. También es interesante remarcar la gran personalización que se puede hacer de dichos componentes, lo que permite seleccionar frecuencias y tiempos de ejecución o conmutación, establecer condiciones iniciales, escoger modos de operación o modelar pérdidas.

Un ejemplo de un diseño realizado con el editor de esquemáticos de Typhoon HIL Control Center aparece representado en la Figura 3-11.

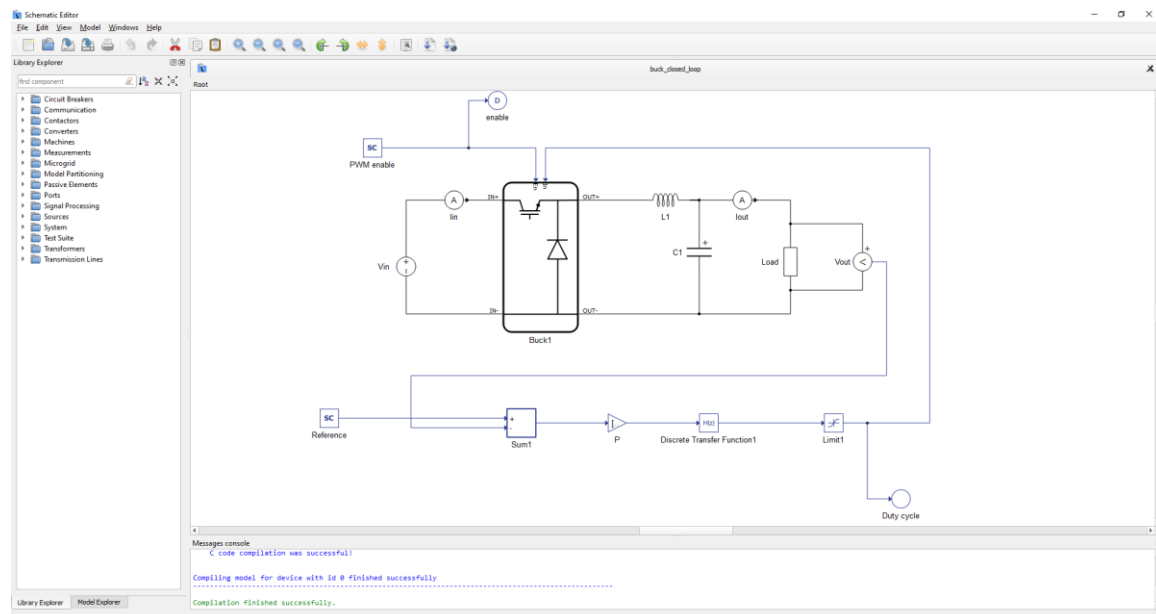


Figura 3-11: Esquemático realizado con Schematic Editor Typhoon HIL Control Center

- **HIL SCADA:**

HIL SCADA se trata de un entorno gráfico muy completo que permite la creación de una interfaz para el testeo del circuito previamente diseñado y compilado mediante el editor de esquemáticos. Es un módulo muy útil ya que permite el despliegue del circuito sobre el *hardware* físico o virtual de simulación y la monitorización de las diferentes señales a tiempo real de forma visual, a través del empleo de diferentes UIs (*User Interfaces*) totalmente personalizables y reprogramables mediante lenguaje Python.

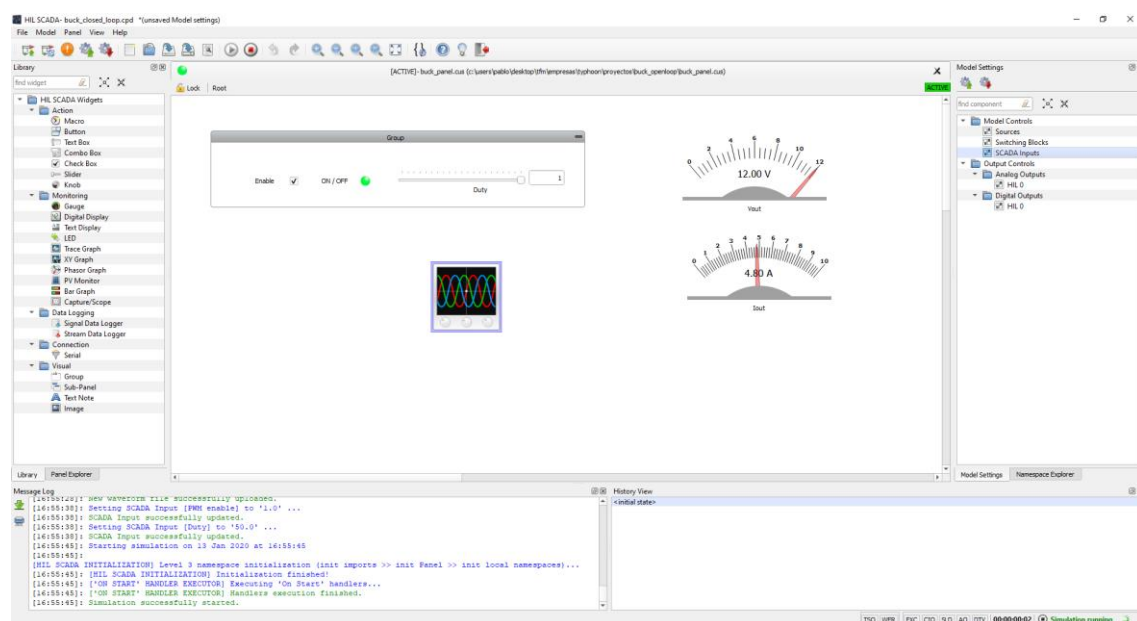


Figura 3-12: Simulación HIL con HIL SCADA Typhoon HIL Control Center

Un ejemplo de simulación HIL corriendo a tiempo real y controlada a través del módulo HIL SCADA de Typhoon HIL Control Center se puede observar en la Figura 3-12.

- **Typhoon Test IDE:**

Este módulo se trata de un entorno de programación que permite al usuario la escritura y ejecución a tiempo real de pruebas automatizadas mediante el empleo de lenguaje Python. Cuenta además con una gran variedad de bibliotecas que facilitan todo el proceso.

Cabe destacar que en este trabajo no se ha hecho hincapié en este módulo ya que se ha creído conveniente seguir el enfoque de simulación mediante SCADA, más gráfico e innovador. Un ejemplo del aspecto de un *script* cargado con esta herramienta se muestra en la Figura 3-13.

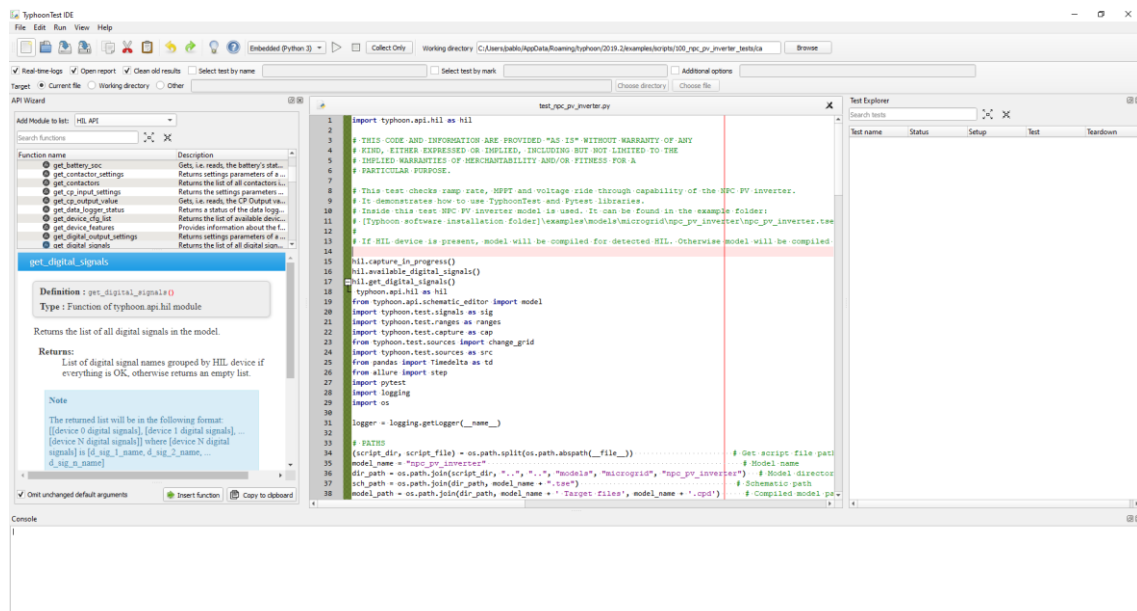


Figura 3-13: Script en Typhoon Test IDE Typhoon HIL Control Center

- **Test Suite:**

Test Suite se trata de una herramienta útil para llevar a cabo una serie de bancos de pruebas comerciales para determinar si el diseño realizado cumple con una serie de normativas para que éste pueda ser certificado.

Durante el desarrollo de este trabajo este módulo aún se encontraba en pruebas y no se ha creído de utilidad para la comparación con otras soluciones comerciales. La Figura 3-14 muestra un ejemplo de cómo sería un proceso de testeo con esta herramienta.

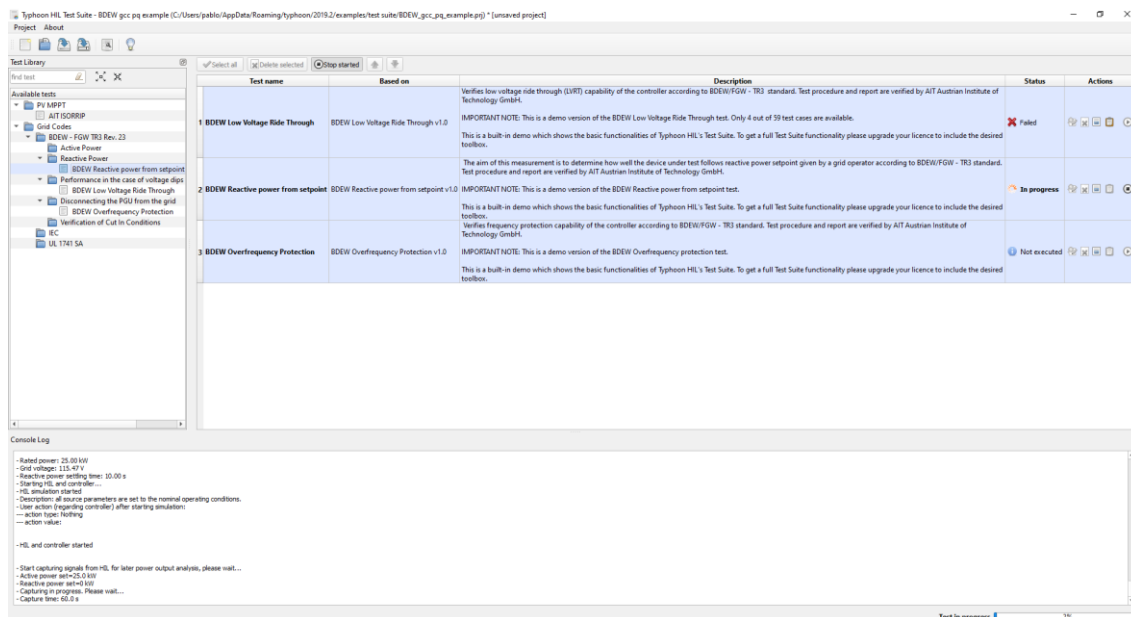


Figura 3-14: Banco de pruebas Test Suite Typhoon HIL Control Center

En los apartados 4.2.2 y 4.3.2, se profundizará con mayor detalle en la configuración, utilización e interconexión de cada uno de los módulos de este programa para el despliegue y simulación de un sistema concreto de cara a poder llevar a cabo la comparativa de resultados obtenidos. El sistema que se diseñará será un convertidor conmutado reductor o *buck*, cuya configuración y parámetros principales se vieron en el punto 2.3 y cuyas particularidades de diseño se comentarán en el apartado 4.1.

Después de haber estudiado las características y módulos principales del programa Typhoon HIL se va a pasar a listar una serie de opciones de las que dispone y peculiaridades que servirán para realizar la comparativa frente a otras herramientas en el mercado.

- **Solvers:**

Typhoon HIL ofrece tres métodos de discretización: exacto, trapezoidal y Euler. Estos modos se pueden ver en la Figura 3-15.

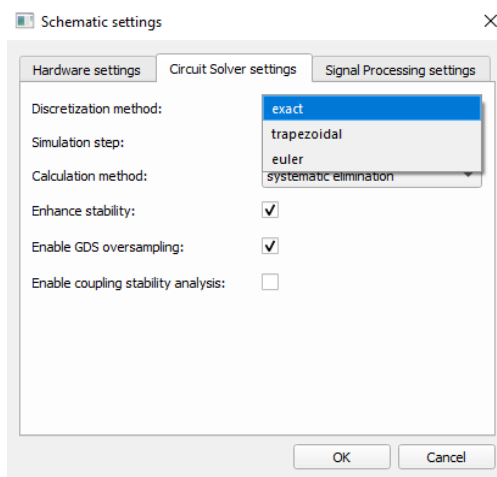


Figura 3-15: Tipos de solvers de Typhoon HIL402

- **RT:**

La simulación del esquemático realizado se lleva a cabo a tiempo real sobre el *hardware* seleccionado. En el caso de este trabajo sobre el HIL402 al disponer de licencia de uso de este en modo máquina virtual. En la Figura 3-16 se puede observar para una máquina HIL determinada las diferentes configuraciones de las que dispone Typhoon HIL.

Device Configuration Table

HIL Device: HIL402

	Configuration 1	Configuration 2	Configuration 3	Configuration 4	Configuration 5	Configuration 6
Number of SPCs	3	4	2	2	2	2
Machine solvers	1	0	0	1	2	1
Signal generators	12	12	12	12	12	12
Look Up Tables	8	8	8	8	8	8
PWM channels	12	12	12	12	12	12
SPC peak processing power [GMACS]	0.6	0.6	1.2	0.6	0.6	0.6
SPC matrix memory [KWords]	16.0	16.0	16.0	16.0	16.0	16.0
SPC output memory size [variables]	256	256	512	512	256	256
Max converter weight (ideal switches) / SPC	3	3	3	3	3	3
Contactors (ideal switches) / SPC	6	6	6	6	6	6
Non-ideal switches / SPC	0	0	32	32	0	0
Time varying elements / SPC	0	0	16	16	0	0
GDS oversampling	yes	yes	yes	yes	yes	yes
Nonlinear machine support	no	no	no	no	no	yes
Switching delay	yes	yes	yes	yes	yes	yes
Converter power loss calculation	no	no	no	yes	no	no

Close

Figura 3-16: Modos de configuración HIL402

- **Modelado de pérdidas:**

Al tratarse el editor de esquemáticos de una herramienta de diseño de circuitos gráfica, en la que se pueden arrastrar y soltar los componentes y bloques e interconectarlos para lograr el resultado buscado, el modelado de pérdidas se puede llevar a cabo de una forma sencilla mediante la incorporación de resistencias de pérdidas junto a los elementos deseados. Esto se puede comprobar en la Figura 3-17.

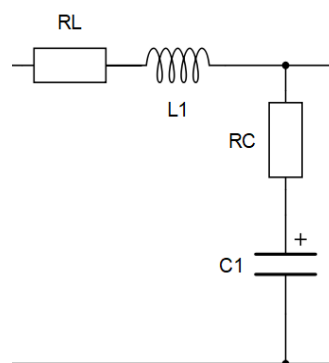


Figura 3-17: Incorporación de pérdidas al circuito Schematic Editor Typhoon HIL

- **Área ocupada:**

Una vez compilado el circuito diseñado y cargado en HIL SCADA, en la consola de mensaje nos aparece un reporte con gran cantidad de información acerca de la utilización de señales, componentes, IO o *Look Up Tables* (LUTs). Un ejemplo de este tipo de reportes aparece en la Figura 3-18.

```

Messages console
Running Device specific hw utilization analysis:
Standard processing core utilization: 1 out of 3 33.33%
Signal generator utilization: 1 out of 12 8.33%
Look up tables utilization: 0 out of 8 0.0%
Machine solver utilization: 0 out of 1 0.0%
Parallel DTV Conv. Detectors utilization: 0 out of 0 0.0%
PWM channels utilization: 1 out of 12 8.33%

Running core0 specific hardware utilization analysis:
Power Electronics Converters utilization: 1 out of 3 33.33%
Contactor utilization: 0 out of 6 0.0%
TVE solvers utilization: 0 out of 0 0.0%
SP sources utilization: 0 out of 16 0.0%
Delayed controlled sources utilization: 0 out of 12 0.0%
Non-Ideal switches utilization: 0 out of 0 0.0%

Starting code generation for system CPU
Nothing to generate.

Starting code generation for user CPU
Signal processing IO variables utilization: 2 out of 4032
Signal processing Probes utilization: 1 out of 512
Signal processing Digital Probes utilization: 1 out of 128
C code generated successfully!
Starting compilation of generated C code...
Total utilization of the internal memory: 46 out of 255 kB (18.26%).
Code segment size: 33 out of 255 kB (13.08%)
Data segment size: 13 out of 255 kB (5.17%)
C code compilation was successful!

Compiling model for device with id 0 finished successfully
-----
Compilation finished successfully.

```

Figura 3-18: Reporte utilización Typhoon HIL

- **Adquisición de datos:**

Una vez realizada una simulación, las señales se pueden monitorizar y capturar a través de la interfaz gráfica de HIL SCADA, gracias al osciloscopio integrado del que dispone HIL402. Luego, es posible exportar y volcar los datos de estas señales en varios formatos: CSV, HDF5, MAT y TDMS.

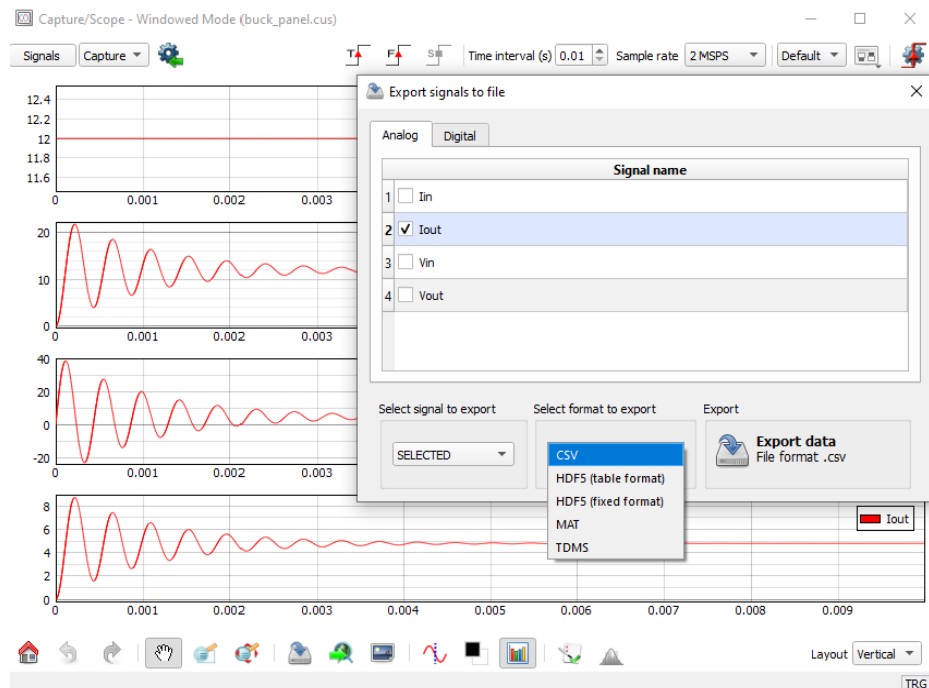


Figura 3-19: Adquisición de datos y formatos Typhoon HIL

Un ejemplo de la adquisición de datos de una señal seleccionada aparece en la Figura 3-19.

- ***On-line / Off-line:***

La simulación realizada es una simulación *Hardware-In-the-Loop* y *On-line*.

- **Sistema operativo:**

El sistema operativo sobre el que corre el *software* Typhoon HIL Control Center es Windows.

3.2 National Instruments

National Instruments [12] se trata de una empresa americana fundada en 1976 con sede en Austin, Texas, y con alrededor de 7500 empleados, dedicada principalmente al desarrollo y venta tanto de *software* como de plataformas *hardware* para el campo de la ingeniería. Sus equipos están centrados fundamentalmente en varias áreas de la electrónica como son: la instrumentación virtual, la adquisición de datos y control o los sistemas de testeo automatizados.



Figura 3-20: Logotipo National Instruments [12]

Es una compañía prestigiosa y ampliamente reconocida en todo el sector de la electrónica cuyo producto estrella es sin duda el entorno de desarrollo gráfico LabVIEW, aunque existen más productos entre la amplia gama ofertada por NI, que abarca desde *software* para el diseño específico de circuitos (Multisim) o PCBs (Ultiboard) hasta controladores industriales a tiempo real (cRIO) o sistemas de adquisición de datos (DAQ).

Entre las aplicaciones principales de los productos de National Instruments destacan la electrónica de potencia, los sistemas de control y las simulaciones a tiempo real y *Hardware-In-the-Loop*. Aunque, como es evidente ante tal variedad de productos, abarcan prácticamente todos los sectores de la electrónica.



Figura 3-21: Logotipo LabVIEW [12]

En este trabajo se profundizará en aquellas herramientas de NI relacionadas con RT y HIL, a fin de conseguir realizar un diseño de un circuito determinado, compilarlo y simularlo sobre un *hardware* a tiempo real, para observar tanto la salida obtenida como los parámetros de utilización y rendimiento.

De este modo, se comentarán primero las bases de LabVIEW, que es el *software* de diseño utilizado. Posteriormente se verán las características del *hardware* NI myRIO-1900 que es el sistema compacto sobre el cual se ha llevado a cabo la simulación y, finalmente se verán dos de los módulos de LabVIEW que permiten llevar a cabo este tipo de simulaciones: LabVIEW Control Design and Simulation (con Multisim) y LabVIEW FPGA Module.

LabVIEW es un *software* propietario de National Instruments (NI) que consiste en un entorno integrado de desarrollo gráfico que permite diseñar modelos, realizar tests, controlar sistemas, tomar medidas y conectar e intercambiar datos con un *hardware* determinado.

En este programa se utiliza un lenguaje de programación gráfico y de relativa sencillez de uso y, además, viene con un gran número de bibliotecas, funciones y bloques predefinidos. Todo esto sumado a la gran cantidad de tutoriales, programas de ejemplo y documentación disponible, hace que LabVIEW no solo se trate de un programa perfecto para su uso en aplicaciones industriales, sino que también es muy recomendable y empleado en los campos de la investigación y de la docencia.

El diseño mediante LabVIEW se estructura en torno a un tipo de archivo básico que lleva por nombre *Virtual Instrument* o VI. Estos archivos llevan la extensión “.vi” y, como ejemplo, serían lo equivalente en MATLAB a un archivo “.m”. Cada VI además puede agruparse y estructurarse en proyectos, donde además se puede definir el *hardware* conectado al que se bajará el diseño. En la Figura 3-22 se puede observar un ejemplo del aspecto que tiene el explorador de proyectos de LabVIEW con la máquina myRIO-1900 conectada y varios VIs incluidos.

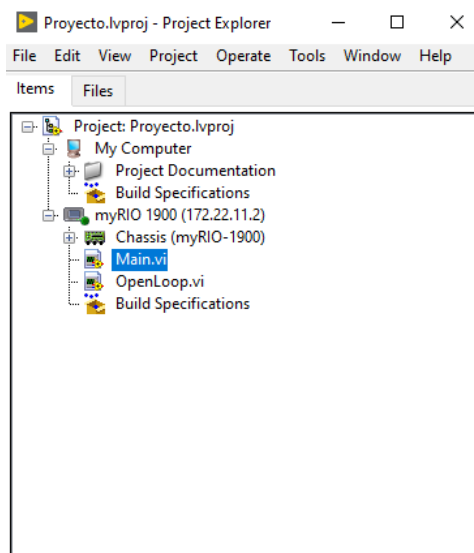


Figura 3-22: Explorador de proyectos LabVIEW

Cada uno de estos VIs está formado por dos elementos o piezas fundamentales, entre las que se puede alternar en cualquier momento (CTRL+E):

- **Front Panel:**

El panel frontal es la ventana del VI usada como interfaz de usuario. Cabe destacar que LabVIEW es un programa con un diseño sencillo y visual, con un enfoque mucho más centrado en la interfaz de usuario, desde la cual se puede observar el comportamiento de cada variable y señal de salida, y también es posible dar nuevos valores a las diferentes entradas.

El panel frontal, que tiene color gris y se encuentra cuadriculado, lo conforman dos elementos: los controles y los indicadores.

Los controles son básicamente las entradas modificables del programa y los hay de numerosos tipos como botones, cuadros para teclear datos, menús desplegables con selección, barras, ruedecillas, etc. En cambio, los indicadores serían las salidas del programa y no se puede alterar su valor desde la interfaz una vez ejecutado. Pueden ser gráficas, LEDs, cuadros con datos, etc.

En la Figura 3-23 aparece un ejemplo de un panel frontal de LabVIEW. En este caso los controles serían el botón para hacer *zoom* a la señal, el botón para parar la simulación y el desplegable para elegir el tipo de medida, mientras que el indicador en este caso sería la gráfica de salida con la señal obtenida.

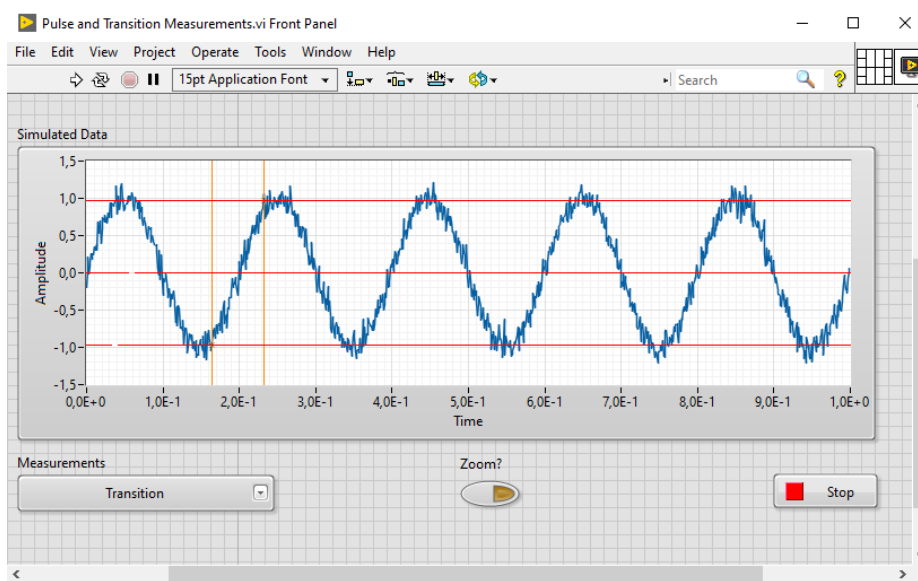


Figura 3-23: Ejemplo panel frontal LabVIEW

- **Block Diagram:**

El diagrama de bloques es el lugar donde se realiza el diseño de la funcionalidad del sistema. Su funcionamiento se basa fundamentalmente en la incorporación e interconexión de bloques y señales dentro de estructuras tipo case o bucles tipo *while* o *for*, hasta lograr el comportamiento deseado. Es decir, que las señales fluyen desde las entradas introducidas en los controles del panel frontal, que es la interfaz de usuario, pasando por los diferentes bloques del diagrama de bloques a través de los “cables” que los conectan, hasta desembocar en algún indicador de salida, para poder analizar los resultados desde el panel frontal.

El diagrama de bloques del panel frontal de ejemplo presentado anteriormente (Figura 3-23) aparece en la Figura 3-24.

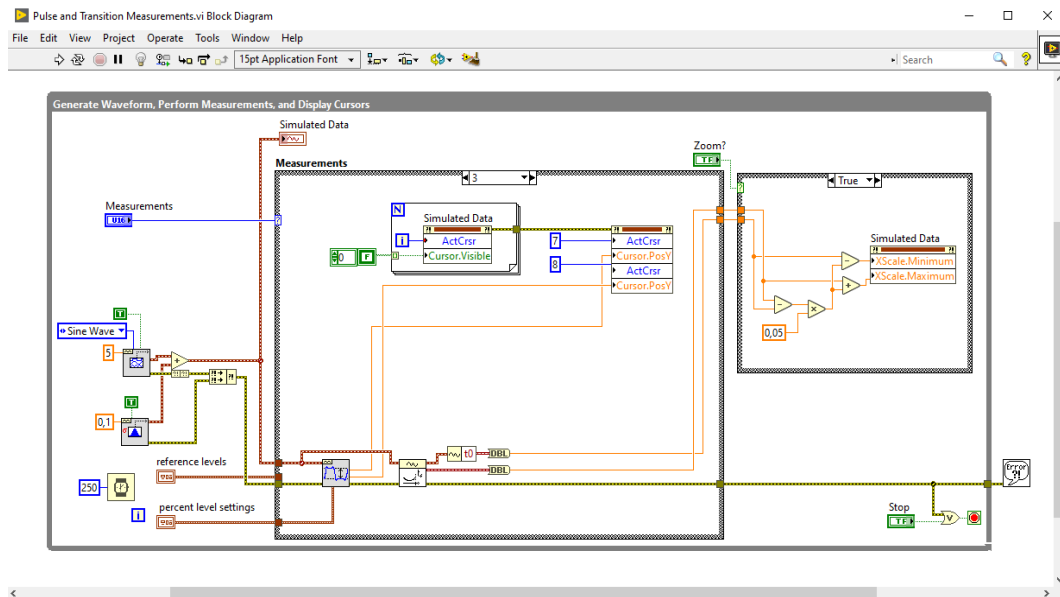


Figura 3-24: Ejemplo diagrama de bloques LabVIEW

Si bien es cierto que este tipo de diseño visual gráfico es bastante intuitivo y sencillo de emplear para una gran variedad de proyectos y modelos, LabVIEW no deja de estar basado en el análisis y la resolución de los sistemas a través de fórmulas matemáticas ejecutándose dentro de bucles y demás estructuras. Como se estudiará en los apartados 3.2.2, 3.2.3 y especialmente en el 4.2.4, el diseño de un circuito mediante este tipo de programación no es para nada trivial, especialmente para un circuito con cierto nivel de complejidad, como puede ser la programación de un convertidor conmutado a través de las ecuaciones que lo modelan. Esto, como se comentará en la comparativa frente a otras soluciones comerciales (3.4), se trata de una desventaja que posee LabVIEW frente a las herramientas de la competencia como Typhoon HIL, donde el diseño de este tipo de circuitos es prácticamente inmediato.

3.2.1 NI myRIO-1900

Para la comparativa entre herramientas, para el caso de National Instruments se parte del sistema embebido NI myRIO-1900 [13] que, si bien es cierto que en la introducción al apartado 3 se comentaba que no se iba a emplear *hardware* físico en los distintos modelos debido al alto coste de las máquinas HIL, en este caso se optó por su empleo por varios motivos:

- El precio de este dispositivo es considerablemente inferior al de una máquina HIL, al no ser específico de este tipo de simulaciones. La placa myRIO-1900 se puede encontrar actualmente en el mercado a unos 572 €, frente a los casi 10.000 € que cuesta una máquina de la serie 4 de Typhoon como es el HIL402.
- Ya se disponía de este dispositivo con anterioridad a la realización de este trabajo, por lo que no supone costes adicionales.
- LabVIEW no permite mostrar resultados de *hardware* sin bajar los modelos a un dispositivo específico.

Por todo ello, se decidió emplear este sistema, que lleva integrado la FPGA de Xilinx Zynq-7000 y cuyo aspecto se puede observar en la Figura 3-25.



Figura 3-25: Dispositivo MyRIO-1900 con FPGA Xilinx Zynq-7000 [14]

Se trata de un sistema embebido que entre sus aplicaciones principales destaca la robótica, la electrónica de potencia y los sistemas de control. Este tipo de dispositivos son especialmente interesantes en el ámbito de la docencia debido a su relativa sencillez de uso y dispone de múltiples entradas/salidas reconfigurables.

Entre sus características principales destacan:

- Sistema Xilinx Zynq integrado en un chip.
- Procesador *dual-core* ARM Cortex-A9.
- Entradas analógicas (10 canales).
- Salidas analógicas (6 canales).

- Entradas y salidas analógicas disponibles a través de conectores de audio Jack de 3,5 mm.
- 40 líneas E/S digitales con posibilidad de acceso inalámbrico.
- Periféricos varios: acelerómetro, LEDs y botón.
- Rango de voltaje de 6 V a 16 V, 14 W de requisito de potencia.
- Conectores de expansión MXP y MSP.
- Compatibilidad *software*: LabVIEW (módulo myRIO).
- Conectividad mediante USB.

3.2.2 LabVIEW Control Design and Simulation Module & Multisim

En cuanto al *software* específico a emplear para la simulación a tiempo real, en el caso de National Instruments, se escogió su herramienta de diseño por excelencia: LabVIEW.

Dada la estructura de funcionamiento propia de LabVIEW que se comentó al final de la introducción del apartado 3.2, es decir, el modelado de sistemas a través de las fórmulas matemáticas, bucles y estructuras que simulen los mismos, se comenzó buscando algún tipo de módulo capaz de lanzar todas estas simulaciones a través del modelo circuital directamente. Esta decisión inicial se tomó de cara a encontrar el procedimiento más sencillo para que posibles usuarios interesados en iniciarse en el campo de las simulaciones HIL, sin experiencia previa, como bien pueden ser estudiantes, tan solo tuvieran que diseñar sus circuitos y simularlos a tiempo real, sin entrar en profundidad de detalles.

Ante esta situación, se decidió emplear en un principio el módulo “Control Design and Simulation” de LabVIEW, junto con el programa Multisim [15].



Figura 3-26: Logotipo Multisim [15]

Multisim se trata de un programa, también de National Instruments, dedicado al diseño y simulación tipo SPICE de circuitos integrados de electrónica de potencia. La idea inicial consistía en lograr integrar un bloque modelado a través de dicha herramienta en un proyecto de LabVIEW, de modo que toda la parte de la lógica interna del circuito se hiciera, con una dificultad considerablemente menor, a través de Multisim y, desde LabVIEW, únicamente se conectarán las entradas y salidas de dicho bloque, que funcionaría a modo de “caja negra”.

Los detalles específicos de un montaje concreto sobre el circuito *buck* seleccionado, se expondrán en el apartado 4.2.3, y se analizarán los motivos por los cuales se ha optado finalmente por el empleo del módulo de FPGA de LabVIEW. Pero, de forma resumida, tiene que ver con las frecuencias con las que se puede simular de este modo a tiempo real, que es de tan solo 1 kHz y, además, no se permite desplegar sobre un dispositivo *hardware* concreto para visualizar los detalles de utilización del modelo.

Las características del empleo de la herramienta LabVIEW junto a Multisim se detallan a continuación:

- **Solvers:**

Si se emplea la simulación a través del módulo “Control Design and Simulation”, es posible establecer el *solver* a emplear entre una gran variedad como Runge-Kutta, Euler, Radau, BDF, etc. como se muestra en la Figura 3-27.

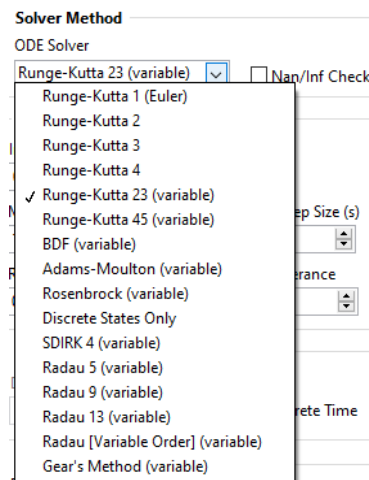


Figura 3-27: Tipos de solvers de LabVIEW Control & Simulation Loop

- **RT:**

El bucle "Control & Simulation Loop" permite simular a varias frecuencias (sobre un PC con relojes de 1 kHz o derivados y con un HW de NI de un 1 MHz o similares), pero para que sea una simulación a tiempo real de verdad, solo se permite, como máximo, 1 kHz con pasos o *steps* de periodos 0,001 s (1 ms), como se verá en el apartado 4.2.3 (Figura 4-28).

- **Modelado de pérdidas:**

En el caso del módulo “Control Design and Simulation”, al haber realizado el diseño del circuito a través de un programa de tipo SPICE como Multisim, la introducción de pérdidas al mismo es prácticamente inmediata. Se podría hacer tanto incluyendo componentes ideales e introduciendo los componentes que simulen las pérdidas (como resistencias de pérdidas), o directamente en las características de cada componente a través de la pestaña *Fault*, como aparece en la Figura 3-28 para el caso de un condensador.

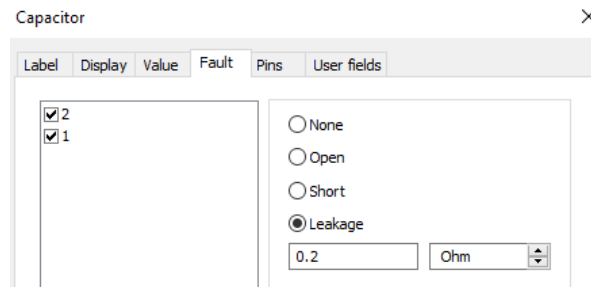


Figura 3-28: Ejemplo pérdidas en condensador Multisim

- **Área ocupada:**

En cuanto al área ocupada, el módulo “Control Design and Simulation” no es capaz de dar una estimación de la utilización de componentes, aunque se despliegue sobre un *hardware* concreto. Esto, junto a las bajas frecuencias soportadas son los motivos principales por los que se empleó LabVIEW FPGA, pese a su mayor complejidad a la hora del diseño.

- **Adquisición de datos:**

En una gráfica obtenida en el módulo “Control Design and Simulation” sí sería posible exportar directamente los resultados o bien al *clipboard* o directamente sobre una hoja de Excel.

- **On-line / Off-line:**

La simulación realizada se trata de una simulación Off-line.

- **Sistema operativo:**

LabVIEW se encuentran disponibles en Windows, Linux y Mac, pero Multisim actualmente solo está disponible para Windows.

3.2.3 LabVIEW FPGA Module

LabVIEW FPGA Module [16] se trata de un *add-on* del programa LabVIEW centrado principalmente en el diseño de sistemas integrados a nivel de FPGA. Por ello, es un módulo idóneo para realizar diseños altamente eficientes y posteriormente simularlos a tiempo real en simulaciones *Hardware-In-the-Loop* sobre un *hardware* específico para observar su rendimiento y la utilización en cuanto a componentes.

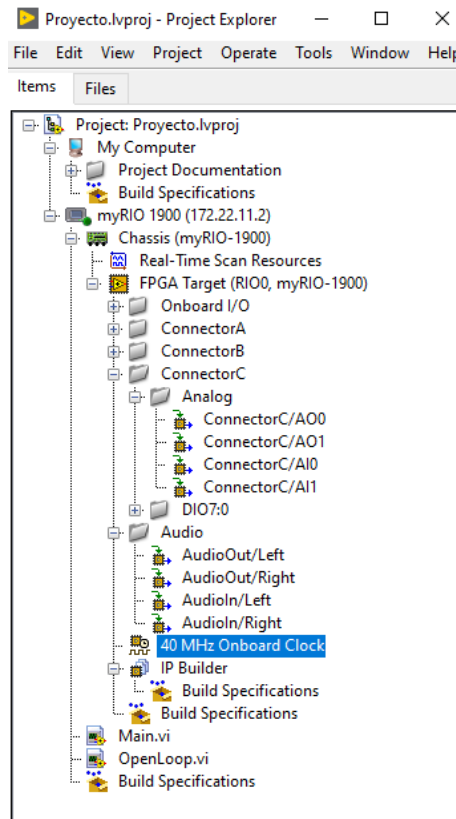


Figura 3-29: Vista de proyecto LabVIEW FPGA

Como se muestra en la Figura 3-29, al crear un proyecto de este tipo y seleccionar un *hardware* en concreto, es posible visualizar todas las entradas y salidas de los diversos conectores de la FPGA. También se puede observar el reloj que viene montado en la placa que, en el caso del dispositivo myRIO-1900, es de 40 MHz. Una característica interesante a la hora del diseño es que es posible arrastrar directamente al diagrama de bloques cualquiera de estos conectores de entrada/salida tanto digitales como analógicos y demás componentes, para realizar las interconexiones pertinentes de manera sencilla. Otra opción, de equivalente resultado, consiste en el empleo de bloques propios de myRIO como se muestra en la Figura 3-30 para sacar las señales de los diferentes conectores del sistema o interactuar con los distintos periféricos. Un ejemplo se puede visualizar en la Figura 3-31 sobre un bloque de salida analógica declarado en el conector C.

El modelado para FPGA como tal es similar en cuanto a proceder respecto a lo mostrado en la introducción al apartado 3.2 y en el 3.2.2, pero en este caso es necesario emplear puertas lógicas y componentes de bajo nivel, así como bucles y estructuras soportados por el *hardware* sobre el que simular. El diseño específico de un circuito con esta herramienta se detallará en el apartado 4.2.4.

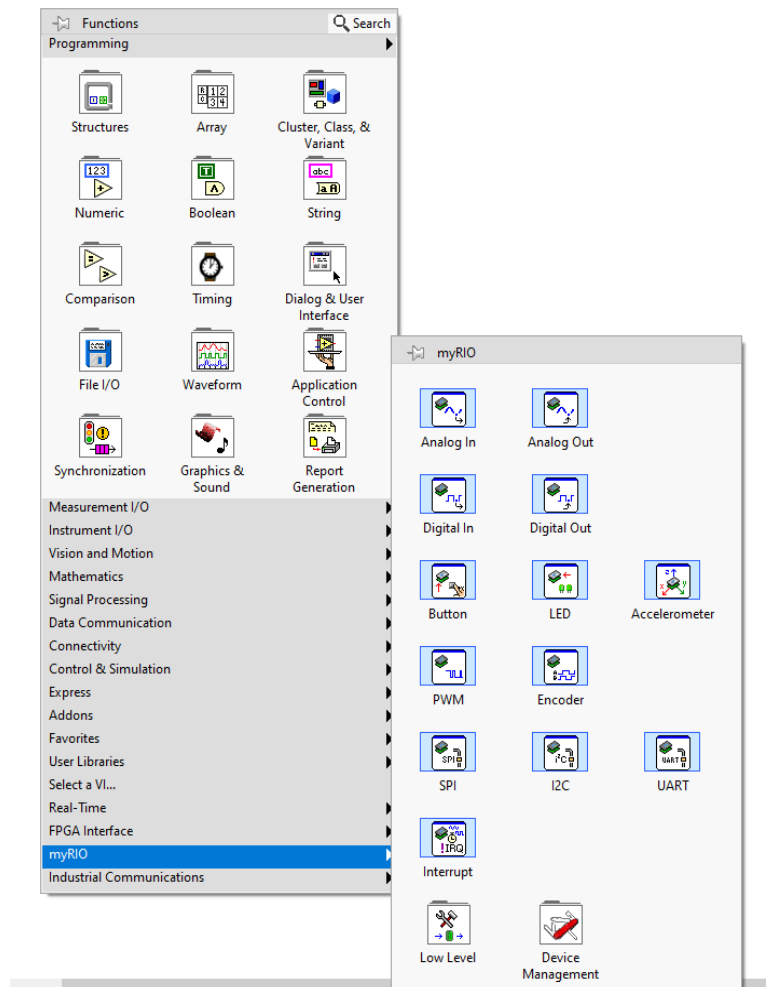


Figura 3-30: Bloques específicos myRIO

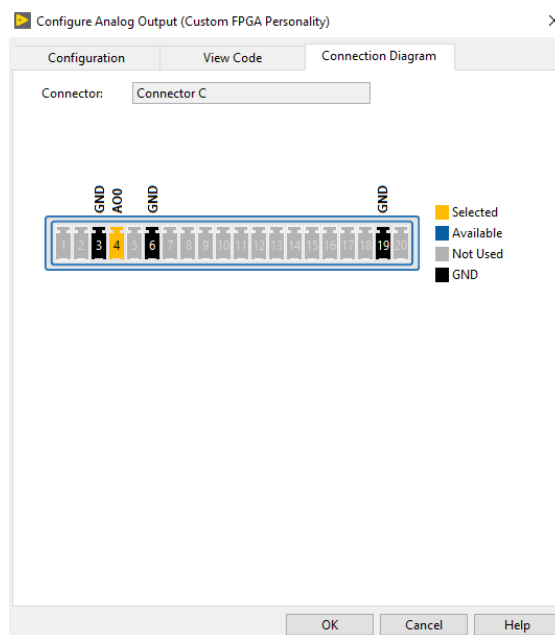


Figura 3-31: Ejemplo de salida analógica conector C myRIO

Tras adquirir una visión general del programa de diseño LabVIEW y su módulo de FPGA, se va a pasar a listar una serie de opciones de las que dispone de forma similar al listado de características que se realizó en el apartado 3.1.2 para Typhoon HIL y en el 3.2.2 para Multisim, de modo que se pueda establecer una comparativa.

- **Solvers:**

En el caso de LabVIEW FPGA, que es sobre el que se realizarán las simulaciones, no es posible declarar un *solver* específico al tener que hacer el diseño a través de puertas lógicas y bucles que simulen una serie de fórmulas matemáticas concretas, que serán éstas las que lleven ya el método específico. En el apartado 4.2.4 se verá cómo se ha usado el método Euler.

- **RT:**

En el caso de LabVIEW FPGA es posible realizar simulaciones a tiempo real sobre un *hardware* concreto como es myRIO-1900 con, en principio, menos restricciones respecto a tiempos y frecuencias. Todo dependerá de la optimización del diseño concreto realizado y de las características del reloj y componentes de la placa sobre la que se diseñe.

- **Modelado de pérdidas:**

Nuevamente para el diseño realizado, sobre LabVIEW FPGA, el modelado de pérdidas dependerá de que en el modelo lógico/matemático, se hayan tenido en cuenta desde el principio, para poder incorporarlas en cualquier momento. En el apartado 4.2.4 se verá cómo, en efecto, se consideraron las pérdidas necesarias.

- **Área ocupada:**

Al compilar y bajar un diseño a una placa, LabVIEW FPGA muestra la utilización, tanto estimada en fase de síntesis como la final en la fase de *placement*. En la Figura 3-32 se muestra un ejemplo de un reporte de utilización estimado en esta plataforma.

Status				
Complete				
Reports				
Estimated device utilization (synthesis)				
Device Utilization	Used	Total	Percent	
Slice Registers	11746	35200	33,4	
Slice LUTs	11667	17600	66,3	
Block RAMs	5	60	8,3	
DSP48s	32	80	40,0	

Figura 3-32: Ejemplo reporte utilización LabVIEW FPGA

- **Adquisición de datos:**

Si se emplea LabVIEW FPGA sobre myRIO-1900, no es posible adquirir los datos de forma tan inmediata, ya que este dispositivo no dispone de osciloscopios integrados. En el apartado 3.1.1 se subrayó la importancia de que HIL402 tuviera osciloscopios integrados.

Por todo esto, la forma de adquirir datos será sacando las diferentes señales a visualizar a las salidas de los conectores del *hardware* y observarlas sobre un osciloscopio físico.

- ***On-line / Off-line:***

La simulación realizada es una simulación *Hardware-In-the-Loop* y *On-line*.

- **Sistema operativo:**

LabVIEW se encuentra disponible en Windows, Linux y Mac.

3.3 MATLAB Simulink

La empresa MathWorks [17], con sede principal en Massachusetts, unos 4500 empleados y con oficinas en 16 países, también ofrece la posibilidad de realizar simulaciones a tiempo real a través de su herramienta por excelencia: MATLAB.



Figura 3-33: Logotipo MathWorks [17]

Desde el entorno de simulación Simulink de MATLAB, es posible acceder a dos módulos basados en este tipo de simulaciones: Simulink Real Time y Simulink Real Time Desktop. Cabe destacar que en el análisis final sobre un circuito específico se ha descartado MATLAB como herramienta a comparar, por motivos que se explicarán en los siguientes apartados, pero que se pueden resumir en que: Simulink Real Time es una herramienta de pago y requiere de un *hardware* específico del que no se disponía y Simulink Real Time Desktop permite simular a tiempo real, pero desde el procesador del ordenador que se ejecuta, lo que no permite observar la utilización y el rendimiento en un *hardware* determinado.

3.3.1 Simulink Real Time

Simulink Real Time se trata de un módulo de licencia de pago de Simulink que permite interactuar con una gran variedad de dispositivos *hardware*, para desplegar sobre éstos los modelos previamente diseñados desde Simulink.

Debido a que para emplear este módulo se necesita, por un lado, una licencia de pago y, por otro, un *hardware* determinado sobre el que realizar la simulación y, no se disponía de ninguno de los dos, no se ha podido probar, sin embargo, es interesante tener en cuenta su existencia en el mercado de las simulaciones HIL, ya que Speedgoat, una de las compañías más importantes de este sector, basa las simulaciones de su *hardware* HIL en el empleo de Simulink Real Time [18].

3.3.2 Simulink Real Time Desktop

El módulo Simulink Real Time Desktop es particularmente interesante ya que permite realizar simulaciones a tiempo real sobre el ordenador que se esté ejecutando, sin necesidad de ningún otro dispositivo *hardware* adicional. Obviamente, esta simulación a tiempo real vendrá limitada por la potencia del equipo del que se disponga, pero además existen una serie de limitaciones en cuanto al periodo máximo de muestreo posible que, de forma normal llega hasta 1 kHz y, en modo externo hasta 20 kHz. Modo externo hace referencia a un modo de funcionamiento en el que mediante Simulink Coder, se codifica el circuito diseñado con información de los bloques I/O y se ejecuta en el *kernel* del sistema

operativo, siendo posibles mayores frecuencias [19]. Cabe destacar también que Simulink Real Time Desktop cuenta con una biblioteca de más de 250 bloques I/O para permitir la conexión con una gran cantidad de dispositivos *hardware* comerciales de empresas tales como National Instruments (entre los cuales el dispositivo myRIO-1900 no está incluido) o Advantech [20].

En cuanto al funcionamiento de Simulink Real Time Desktop, está basado fundamentalmente en el empleo de un bloque de sincronismo llamado “Real-Time Sync”, cuyo aspecto se ilustra en la Figura 3-34.

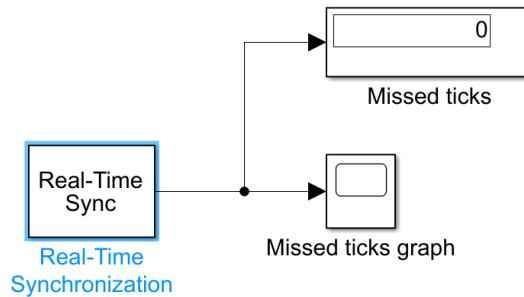


Figura 3-34: Bloque Real-Time Synchronization Simulink Real Time Desktop

Dicho bloque sirve para sincronizar la simulación de modo que se pueda ejecutar en tiempo real. Para ello hay dos parámetros disponibles: por un lado, está el periodo de muestreo que se va a emplear en la simulación y, por otro, el número máximo de *missed ticks* permitido. Este parámetro hace referencia al número de “fallos” permitido durante la simulación en tiempo real ya que, si se escoge un periodo de muestreo muy pequeño, para que Simulink pueda ejecutarlo a tiempo real sobre el ordenador seleccionado, es muy posible que vaya a tener que “saltarse” algunos datos, de modo que logre llegar a simular en el tiempo permitido. Por tanto, se trata de algún modo de la tolerancia del sistema e indica cuán estricto se desea que sean los datos de la simulación a tiempo real. Son por ello, dos valores que se encuentran bastante interrelacionados. La configuración del bloque “Real-Time Synchronization” se puede observar en la Figura 3-35.

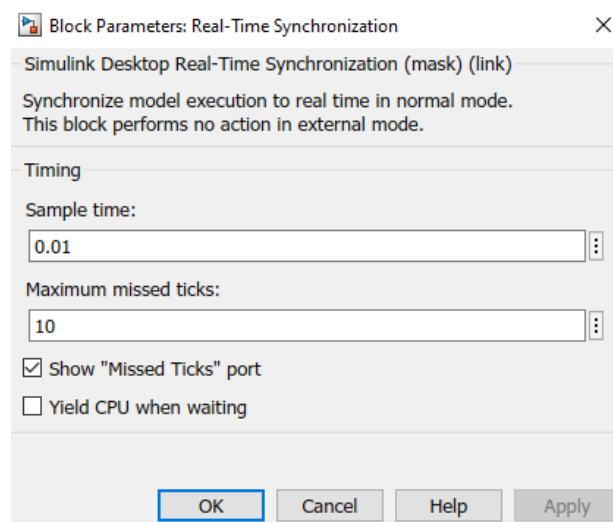


Figura 3-35: Configuración bloque Real-Time Synchronization Simulink Real Time Desktop

Su integración en un diseño de Simulink es muy sencilla. Basta con colocar dicho bloque en la misma hoja “.slx” que el circuito y ya aplica todas las restricciones seleccionadas. En la Figura 3-36 se puede ver su integración en un circuito determinado, que no es otro que el controlador *buck* en lazo abierto que se detallará en el apartado siguiente con mayor profundidad.

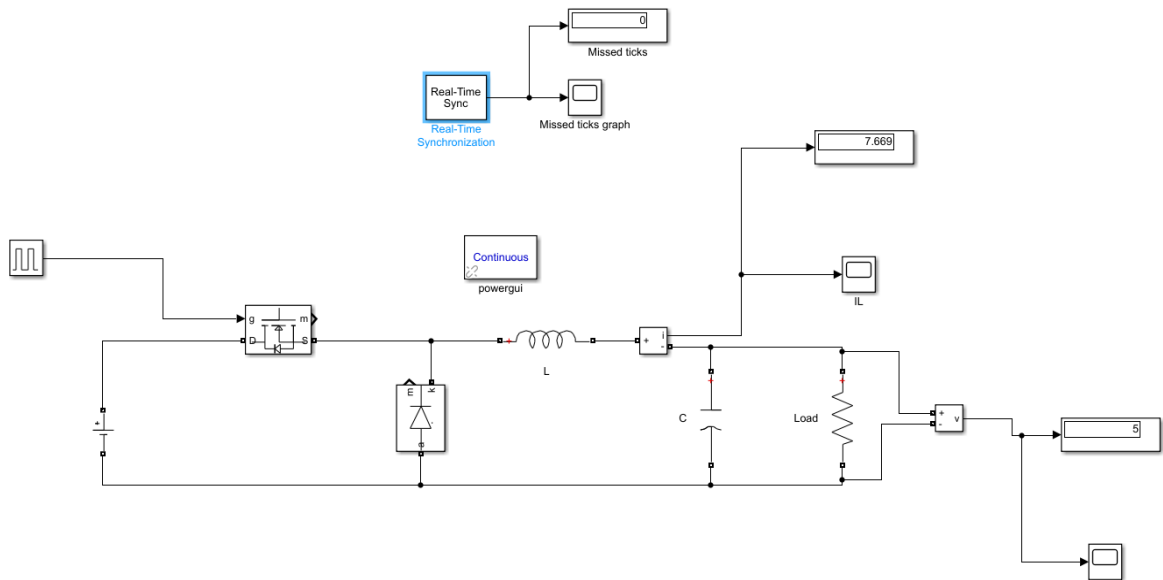


Figura 3-36: Esquema *buck* Simulink Real Time Desktop

Para que el equipo soporte la simulación en tiempo real, debe haber, por tanto, un compromiso entre el periodo de muestreo y los *ticks* permitidos o, de otro modo, la simulación no será posible. Un ejemplo de esto aparece en la Figura 3-37 en la que se ha seleccionado un periodo de muestreo muy pequeño y, se permiten muy pocos *ticks* fallados, es decir, pocos errores.

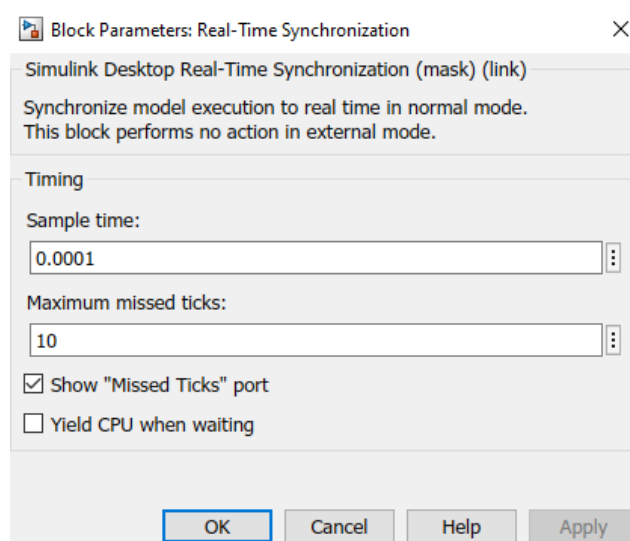


Figura 3-37: Muestreo pequeño Bloque Real-Time Synchronization

Al intentar realizar la simulación aparece una ventana de error en la que se detalla que el número de *ticks* elegido, ha sido excedido y, por tanto, no se puede concluir la simulación, como se muestra en la Figura 3-38.

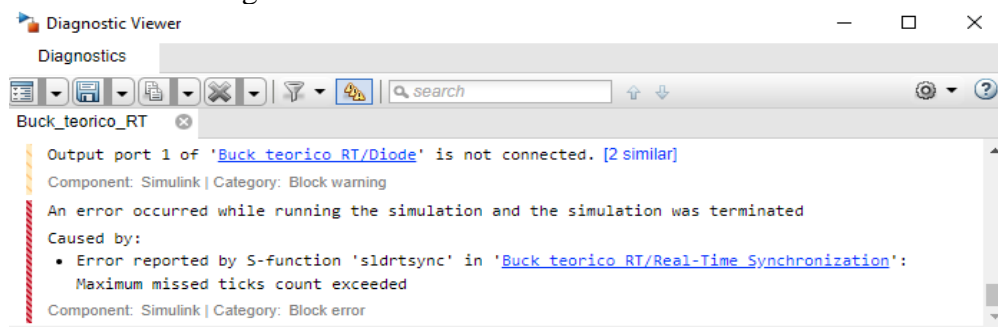


Figura 3-38: Error *ticks* máximos excedidos

Por ello, cabe concluir que, si se desea simular a tiempo real sobre el ordenador en cuestión y con ese periodo de muestreo, será necesario permitir algunos fallos más para que el sincronismo del modelo y el equipo sea posible. La Figura 3-39 ilustra el resultado en cuanto a *ticks* que muestra el bloque sobre una simulación correcta.

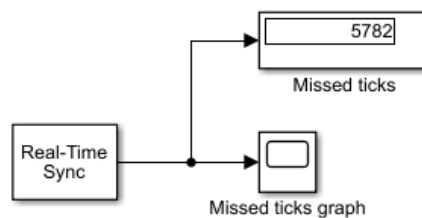


Figura 3-39: *Ticks* obtenidos simulación *buck* Real Time (*time step* = 0,1 ms)

Por último, en la Figura 3-40 se puede observar la evolución de los *ticks* según transcurre el tiempo la misma simulación.

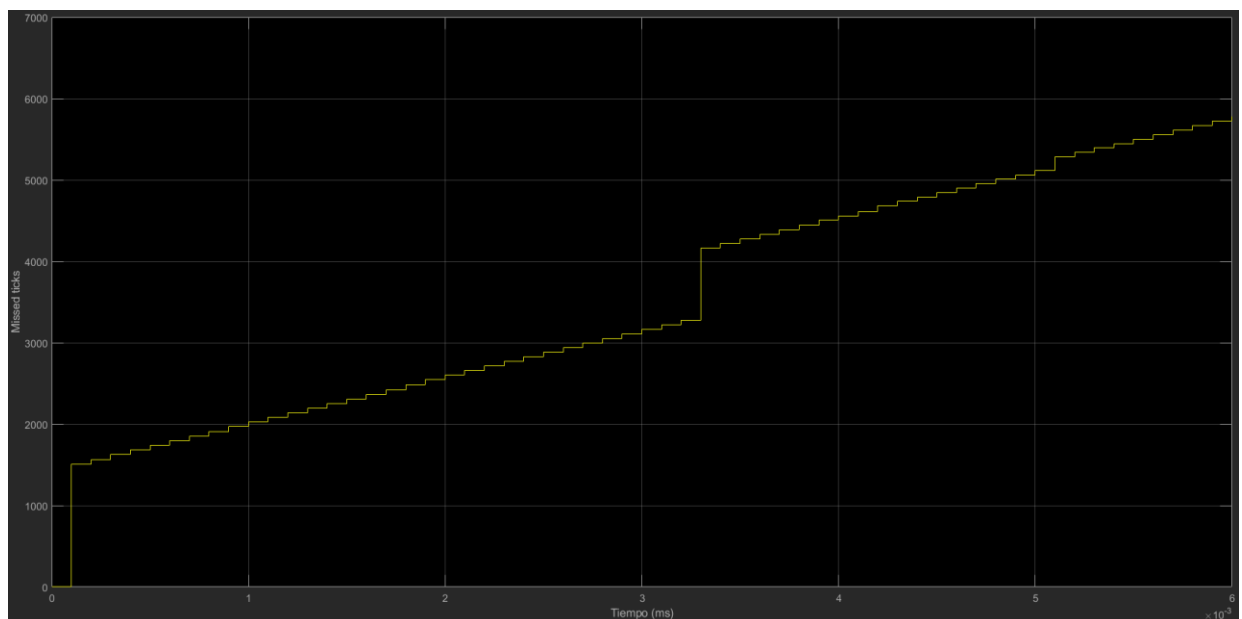


Figura 3-40: Evolución *ticks* con el tiempo

A continuación, se procederá a enumerar las distintas características de Simulink Real Time y Simulink Real Time Desktop, para poder establecer una comparativa con las demás plataformas:

- **Solvers:**

Desde Simulink es posible escoger de entre un conjunto de *solvers* para llevar a cabo la simulación, como Euler, Runge-Kutta o Dormand-Prince. En la Figura 3-41 se muestra el listado de *solvers* disponibles.

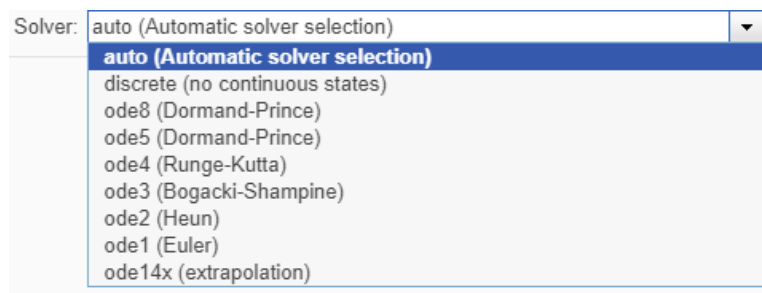


Figura 3-41: Tipos de *solvers* de Simulink

- **RT:**

Simulink Real Time funciona a tiempo real sobre el *hardware* conectado al equipo y enlazado al modelo diseñado.

En el caso de Simulink Real Time Desktop, empleando el bloque “Real-Time Synchronization” hay dos métodos posibles de simulación a tiempo real: por un lado el modo normal, que permite simular hasta frecuencias de 1 kHz y se ejecuta sobre el procesador del propio equipo en el que se ejecuta, y también está el modo externo que traduce el modelo diseñado a un modelo C/C++ lo que permite simular a una mayor frecuencia, de hasta 20 kHz a tiempo real.

- **Modelado de pérdidas:**

Desde el esquemático del circuito que se está diseñando se permite incluir pérdidas de todo tipo en cada componente: resistencias, bobinas, diodos, MOSFET, etc. En la Figura 3-42 se muestra un ejemplo de todas las opciones de pérdidas para el caso concreto de un MOSFET.

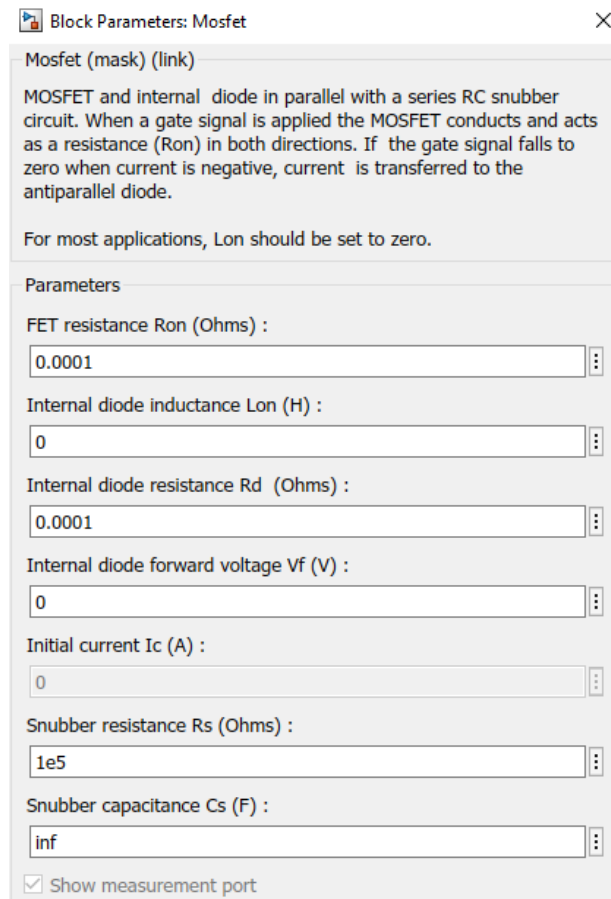


Figura 3-42: Ejemplo pérdidas en MOSFET Simulink

- **Área ocupada:**

Desde Simulink Real Time sería posible examinar la utilización de recursos sobre el *hardware* en el que se realice el despliegue.

En el caso de Simulink Real Time Desktop no se visualiza la utilización dado que no se despliega el modelo sobre un *hardware* concreto, sino que se ejecuta en el propio procesador del PC.

- **Adquisición de datos:**

Se pueden volcar los datos de las distintas señales del circuito a gráficas y, posteriormente, guardar la figura obtenida en la simulación como un archivo de tipo “.fig”, para luego, mediante un sencillo *script*, extraer los datos de las gráficas y guardarlos en un documento de texto.

- **On-line / Off-line:**

La simulación sobre Simulink Real Time sería HIL y On-line, mientras que en el caso de Simulink Real Time Desktop se trata de simulaciones Off-line.

- **Sistema operativo:**

Simulink se encuentra disponible tanto en Windows como para Linux y Mac.

3.4 Comparativa características herramientas comerciales

En este apartado se resumirá la comparativa de características, detalladas en los capítulos anteriores, para las herramientas comerciales estudiadas: Typhoon HIL y LabVIEW de National Instruments. También se incluirán detalles de Simulink Real Time y Simulink Real Time Desktop que, pese a no haberse realizado un despliegue HIL sobre algún *hardware* específico, se creyó conveniente analizar dado que es el punto de partida de simulaciones de este tipo para un gran número de fabricantes. Además de la información de características recopilada en las tablas, se aportarán comentarios tras la familiarización con cada uno de estos, de cara a lograr un análisis más detallado.

- Solvers

Typhoon HIL	LabVIEW		Simulink	
	Control Design & Simulation	LabVIEW FPGA	Real Time Desktop	Real Time
Exact	Runge-Kutta 1 (Euler), 2, 3, 4, 23 (variable), 45 (variable)	Con LabVIEW FPGA el <i>solver</i> utilizado viene definido en el modelo lógico y matemático escogido durante la realización de cada diseño.	Discrete (no continuous states) Ode8 (Dormand-Prince) Ode5 (Dormand-Prince) Ode4 (Runge-Kutta) Ode3 (Bogacki-Shampirne) Ode2 (Heun) Ode1 (Euler) Ode14x (extrapolation)	
Trapezoidal	BDF (variable)			
Euler	Adams-Moulton (variable)			
	Rosenbrock (variable)			
	Discrete States Only			
	SDIRK4 (variable)			
	Radau 5, 9, 13, [Variable Order] (variable)			
	Gear's Method (variable)			

Tabla 3-1: Tabla comparativa *solvers*

- **Adquisición de datos**

Typhoon HIL	LabVIEW		Simulink	
	Control Design & Simulation	LabVIEW FPGA	Real Time Desktop	Real Time
<p>Se permite la exportación en varios formatos de todas las señales capturadas durante la simulación en HIL SCADA:</p> <ul style="list-style-type: none"> - CSV - HDF5 (table format) - HDF5 (fixed format) - MAT - TDMS <p>Cabe destacar que el sistema HIL402 cuenta con osciloscopios integrados, lo que permite hacer dicha captura desde <i>software</i>.</p>	<p>Deja exportar los resultados sacados de una gráfica directamente en EXCEL o copiarlos al <i>clipboard</i>.</p>	<p>Con otro <i>hardware</i> se podrían sacar las señales simuladas a una gráfica o a un fichero para su extracción, pero myRIO no cuenta con osciloscopios integrados y, por lo tanto, es necesario sacar las a salidas del sistema y observarlas con un osciloscopio externo para luego pasarlas a formato de datos para manejar (Excel, CSV...).</p>	<p>Se pueden volcar los datos de las distintas señales a una gráfica y después guardar la figura obtenida en la simulación como un archivo “.fig”, para luego con un sencillo <i>script</i> extraer los datos y guardarlos en un documento de texto.</p>	

Tabla 3-2: Tabla comparativa adquisición de datos

- **Frecuencia de reloj (f_{clk})**

Typhoon HIL	LabVIEW		Simulink	
	Control Design & Simulation	LabVIEW FPGA	Real Time Desktop	Real Time
Con la máquina virtual del <i>hardware</i> HIL402: 50 MHz	Permite simular a varias frecuencias los bucles de “Control Design & Simulation” (1 kHz, 1MHz e intermedios) A tiempo real (RT) la máxima frecuencia permitida es de: 1 kHz	Máxima frecuencia de reloj myRIO-1900 con FPGA Zynq-7000: 40 MHz	Máxima frecuencia posible a tiempo real (RT) en <u>modo normal</u> : 1 kHz En <u>modo externo</u> : 20 kHz	Frecuencias posibles a tiempo real mucho mayores que Real Time Desktop. Depende del <i>hardware</i> a emplear.

Tabla 3-3: Tabla comparativa frecuencia de reloj

- **Frecuencia de conmutación (f_{sw})**

Typhoon HIL	LabVIEW		Simulink	
	Control Design & Simulation	LabVIEW FPGA	Real Time Desktop	Real Time
Con la máquina virtual del <i>hardware</i> HIL402: 200 kHz	La frecuencia de conmutación no será en ningún caso superior que la de reloj: 1 kHz	Depende del <i>hardware</i> a emplear, pero nunca superior a la frecuencia de reloj: 40 MHz	En <u>modo normal</u> : 1 kHz En <u>modo externo</u> es función del <i>Duty</i> seleccionado: $f_{sw} = f_{CLK} \times Duty$ De modo que para el máximo de 20 kHz y con un <i>Duty</i> al 50 % se obtiene una frecuencia de conmutación máxima de: 10 kHz	Frecuencias posibles a tiempo real mucho mayores que Real Time Desktop. Depende del <i>hardware</i> a emplear.

Tabla 3-4: Tabla comparativa frecuencia de conmutación

- **Paso de simulación mínimo (“dt”)**

Typhoon HIL	LabVIEW		Simulink	
	Control Design & Simulation	LabVIEW FPGA	Real Time Desktop	Real Time
Con <i>hardware</i> HIL402: 500 ns	 1 ms	Con un reloj de 40 MHz: 25 ns	En <u>modo normal</u> : 1 ms En <u>modo externo</u> : 50 µs	<i>Steps</i> posibles mucho menores que los de Real Time Desktop. Depende del <i>hardware</i> a emplear.

Tabla 3-5: Tabla comparativa paso mínimo simulación

- **Simulación a Tiempo Real (RT)**

Typhoon HIL	LabVIEW		Simulink	
	Control Design & Simulation	LabVIEW FPGA	Real Time Desktop	Real Time
Funciona en <i>Real Time</i> sobre la máquina virtual HIL402.	El bucle "Control & Simulation Loop" permite simular a varias frecuencias (sobre un PC con relojes de 1 kHz o derivados y con un HW de NI de un 1 MHz o similares), pero para que sea una simulación a tiempo real de verdad, solo se permite 1 kHz con pasos o <i>steps</i> de periodos de 1 ms.	A tiempo real sobre el <i>hardware</i> de NI especificado, en este trabajo: myRIO-1900.	A tiempo real en las frecuencias especificadas. Para lograr esto en modo normal emplea el procesador del propio ordenador en que se ejecuta. Hay que emplear un módulo llamado "Real-Time Synchronization" sobre el que se especifica el periodo de muestreo y el número de <i>Maximum Missed Ticks</i> que indica los errores permitidos con tal de simular el modelo a tiempo real, y que cumpla con los tiempos especificados. En modo externo el modelo del circuito se convierte a un modelo C/C++ lo que implica una mayor frecuencia para simular. En este modo se cometen menos errores a bajas frecuencias (menos <i>missed ticks</i>).	A tiempo real sobre el <i>hardware</i> conectado al equipo y enlazado al modelo diseñado.

Tabla 3-6: Tabla comparativa simulación a tiempo real

- **Modelado de pérdidas**

Typhoon HIL	LabVIEW		Simulink	
	Control Design & Simulation	LabVIEW FPGA	Real Time Desktop	Real Time
En el propio esquemático del modelo circuital se pueden añadir resistencias de pérdidas y variar los valores de bobinas, condensadores, etc. Sin embargo, está algo limitado en algunos bloques que no admiten pérdidas.	Mediante el empleo del programa Multisim, en el diseño del circuito se pueden modelar pérdidas de todo tipo e incluso probar con diferentes modelos de componentes.	La inclusión de pérdidas depende de si se han tenido en cuenta desde el principio en el diseño a bajo nivel del circuito.	En el diseño circuital todos los componentes permiten modelar pérdidas: diodos, MOSFETs, bobinas, condensadores...	

Tabla 3-7: Tabla comparativa modelado de pérdidas

- **Utilización de recursos / área ocupada**

Typhoon HIL	LabVIEW		Simulink	
	Control Design & Simulation	LabVIEW FPGA	Real Time Desktop	Real Time
En el "Schematic Editor" se muestran todos los datos del HW empleado y, una vez que se lanza la simulación, aparece un resumen de los recursos HW, memoria y tamaño de SW. Usa más recursos integrados de alto nivel, como convertidores.	No es sobre un <i>hardware</i> específico, luego no muestra detalles de utilización.	Bajando el diseño a placa se puede observar la utilización tanto estimada, en síntesis, como la final. El reporte se muestra en términos de componentes de bajo nivel (<i>slices</i> , LUTs, DSPs...).	Desplegando sobre un <i>hardware</i> específico mediante Simulink Real Time es posible analizar la utilización de recursos.	

Tabla 3-8: Tabla comparativa utilización de recursos

- **Licencias**

Typhoon HIL	LabVIEW		Simulink	
	Control Design & Simulation	LabVIEW FPGA	Real Time Desktop	Real Time
<p>Registro gratuito y descarga del <i>software</i> "Typhoon HIL Control Center" también gratuito.</p> <p>Se puede usar libremente y se permite emplear la máquina virtual con una licencia de 1 año.</p>	<p>Para usar tanto LabView 2019 como Multisim 14.2 (la versión compatible con LabVIEW 2019) es necesario tener una licencia, o usar la versión de prueba de 45 días que ofrece cada uno.</p> <p>Además, para simular hay que descargar el módulo "Control Design And Simulation".</p>	<p>Para simular, se ha de disponer del módulo NI myRIO incluido con el <i>hardware</i>, y del módulo de pago LabVIEW FPGA.</p>	<p>Con la licencia universitaria de estudiante de Matlab se puede acceder a Simulink Real Time Desktop (no Simulink Real Time).</p> <p>Para realizar los diseños circuitales es necesario descargar el <i>toolkit</i>: "Simscape Electrical", de pago, pero con una versión de prueba de 1 mes.</p>	

Tabla 3-9: Tabla comparativa de licencias

- **On-line / Off-line:**

Typhoon HIL	LabVIEW		Simulink	
	Control Design & Simulation	LabVIEW FPGA	Real Time Desktop	Real Time
<i>On-line</i>	<i>Off-line</i>	<i>On-line</i>	<i>Off-line</i>	<i>On-line</i>

Tabla 3-10: Tabla comparativa online/offline

- **Usabilidad / Documentación:**

Typhoon HIL	LabVIEW		Simulink	
	Control Design & Simulation	LabVIEW FPGA	Real Time Desktop	Real Time
<p>Mucha documentación y gran cantidad de tutoriales en su plataforma Typhoon HIL Academy, con vídeos explicativos y diseños de ejemplo en funcionamiento.</p> <p>Registro en web de ayuda y formación HIL Academy gratis.</p> <p>No hay demasiada ayuda por parte de la comunidad en foros o similares para la resolución de dudas.</p> <p>El diseño es sencillo, tipo esquemático SPICE con muchas opciones para la simulación mediante interfaces gráficas.</p>	<p>Hay gran cantidad de información y ejemplos de LabVIEW al ser un programa de uso muy extendido en el sector. Sin embargo, la mayoría de la información encontrada respecto a lo tratado en el trabajo es algo superficial y poco específica.</p> <p>No es nada trivial montar un circuito (convertidor conmutado, por ejemplo) ya que hay que partir de las ecuaciones matemáticas que lo modelan y, por ello, se partió en un primer momento de Multisim, pero al observar sus limitaciones hubo que hacerlo a bajo nivel, con todas las dificultades que ello conlleva.</p>		<p>Documentación moderada, salvo la guía de uso oficial de "Simulink Real Time Desktop" del propio Matlab, que es completa e incluye diseños a modo de ejemplo sobre los que partir para montar tu propia aplicación a tiempo real.</p> <p>Diseño del esquemático también a modo circuito SPICE, sencillo de montar y simular.</p>	

Tabla 3-11: Tabla comparativa usabilidad/documentación

A partir de la información mostrada en las tablas comparativas anteriores y, a lo largo de todo el capítulo 3, se observa que Typhoon HIL sobre el *hardware* HIL402 tiene una mayor potencia y capacidad que LabVIEW sobre la máquina myRIO-1900, ya que, el dispositivo HIL402 está especialmente diseñado y optimizado para simulaciones del tipo HIL frente a una placa de propósito algo más general como myRIO-1900.

Por otro lado, mientras que la herramienta LabVIEW FPGA parece que permite un control mucho mayor del diseño, algo que se verá con más detalle en un ejemplo práctico en el capítulo siguiente, en términos de frecuencias, pasos de simulación y control de la lógica, las mecánicas de modelado y simulación de Typhoon parecen bastante más sencillas, al ser de tipo esquemático, junto con la posibilidad de simular sobre una máquina virtual, hacen posible desplegar una simulación HIL real en cuestión de minutos.

Typhoon parece que requiere menos esfuerzo para la iniciación en el campo HIL o su empleo por parte de usuarios de todos los niveles de experiencia. Además, la cantidad de ayuda y documentación que se ofrece de manera gratuita a través de la plataforma Typhoon Academy, en especial en todo lo relacionado a la electrónica de potencia, es mayor que la que se puede encontrar sobre LabVIEW en este ámbito. También hay que destacar que la licencia de prueba que ofrece Typhoon de manera gratuita que incluye el uso de la máquina virtual tiene una duración considerablemente mayor, existiendo incluso la posibilidad de alargar este periodo de prueba.

Por otro lado, si bien Typhoon parece más sencillo de emplear, LabVIEW parece bastante más profundo en cuanto a personalización se refiere, ya que, si bien los diseños son más complejos de realizar, también parecen más flexibles en cuanto a las posibilidades de modelado, al estar hechos a través de componentes de bajo nivel, puertas lógicas y bucles, lo cual permitiría exprimir la potencia del sistema al máximo de manera óptima, pero también puede requerir de una mayor experiencia por parte de sus usuarios.

4 Integración, pruebas y resultados

En este apartado se analizarán en profundidad las diferencias entre las herramientas anteriormente expuestas a través del montaje y simulación de un modelo circuital concreto, equivalente para todas ellas, con pérdidas y sin pérdidas. El circuito base a montar será un convertidor conmutado reductor o *buck* asíncrono, con la configuración y características del empleado en mi propio Trabajo de Fin de Grado [1]. La versión ideal o teórica del circuito sobre la que comparar las demás plataformas se diseñará mediante Simulink con el *toolkit* Simscape Electrical, que cuenta con gran cantidad de bloques para la simulación y el diseño de sistemas de electrónica de potencia, donde se forzará la idealidad de todos los componentes (*switches*, diodos, transistores, etc.). Simulink Real Time Desktop no se comparará dado que, como se ha explicado con anterioridad, no se despliega sobre un *hardware* concreto ni muestra detalles de rendimiento/utilización, luego el foco de la comparativa será entre las dos principales herramientas HIL con las que se ha trabajado: Typhoon y LabVIEW.

4.1 Características buck

Las características del convertidor *buck* según lo establecido en mi TFG, y que se corresponden al esquema visto en la Figura 2-5 del apartado 2.3, son las siguientes:

- Voltaje de entrada: $V_S = 12 \text{ V}$
- Voltaje de salida: $V_O = 5 \text{ V}$
- Ciclo de trabajo (*Duty*): $\text{Duty} = V_O / V_S = 5 / 12 = 0,41667$
- Bobina: $L = 22 \mu\text{H}$
- Condensador: $C = 220 \mu\text{F}$
- Carga: $R = 0,625 \Omega$
- Frecuencia de conmutación: $f_{\text{sw}} = 200 \text{ kHz}$
- Periodo de conmutación: $T_{\text{sw}} = 1 / f_{\text{sw}} = 5 \mu\text{s}$

Las pérdidas que se desearán incluir en un principio, antes de comprobar las limitaciones de cada plataforma son las siguientes:

- Resistencia Drain-Source: $R_{\text{DS}_{\text{ON}}} = 0,02 \Omega$
- Resistencia bobina: $R_L = 0,02 \Omega$
- Resistencia equivalente serie: $\text{ESR} = 0,1 \Omega$
- Caída de voltaje diodo: $V_D = 0,4 \text{ V}$
- Resistencia diodo: $R_D = 0,02 \Omega$

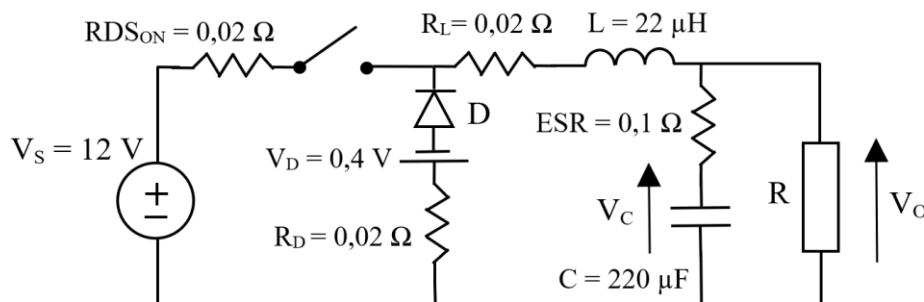


Figura 4-1: Circuito *buck* asíncrono con pérdidas

4.2 Diseño, simulación y montaje convertidor conmutado buck (sin pérdidas)

Para llevar a cabo una simulación equivalente en las plataformas a comparar, hay una serie de características de simulación/diseño que se han tenido que unificar para realizar una adecuada comparación. Evidentemente el valor de cada componente del circuito será el mismo en todos los casos (resistencias, bobinas, condensadores, tensión de entrada...), pero además hay dos características principales a unificar: el paso de simulación o “dt”, que se ajustará al mínimo posible de Typhoon HIL (500 ns), dado que, como se ha visto en el apartado 3.4 es el limitante en este aspecto, luego habrá que ajustar el modelo de LabVIEW a ese paso; y, por otro lado, el tipo de *solver* a emplear en la simulación, se seleccionará Euler en ambos casos, ya que se trata de un *solver* común a las dos herramientas.

4.2.1 Modelo teórico

Para realizar el modelo teórico del convertidor *buck* en Simulink, se ha partido de algunos de los bloques del *toolkit* “Simscape Electrical”, como el MOSFET que hace de interruptor o el diodo. A todos los componentes se les ha dado los valores comentados en el apartado 4.1, en este caso sin la introducción de ningún tipo de pérdidas. En la Figura 4-2 se puede observar el montaje del circuito completo sobre Simulink.

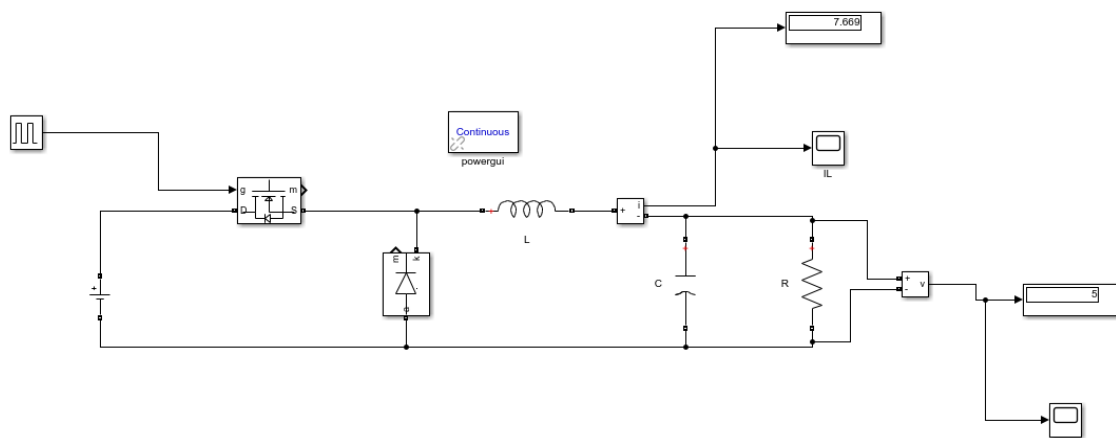


Figura 4-2: Simulación *buck* teórico Simulink (sin pérdidas)

A la entrada del MOSFET para poder realizar la función de *switching*, se introduce una señal cuadrada PWM de amplitud unidad con el periodo de conmutación seleccionado y el *Duty* escogido para obtener 5 V a la salida con 12 V de entrada.

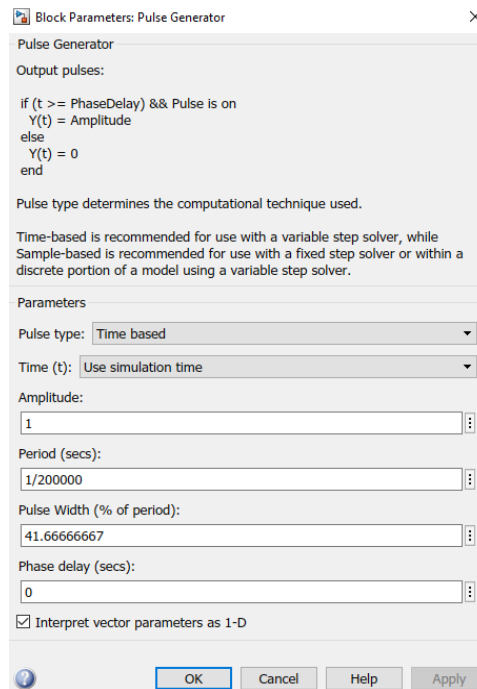


Figura 4-3: Generación PWM *buck* Simulink

En serie a la bobina se coloca un amperímetro para medir la corriente que pasa por ella (I_L) y paralelamente a la carga se coloca un voltímetro para obtener la tensión de salida (V_O). Ambas medidas se conectan a sendos visores para comprobar los resultados y se vuelcan a gráficas para conseguir su representación.

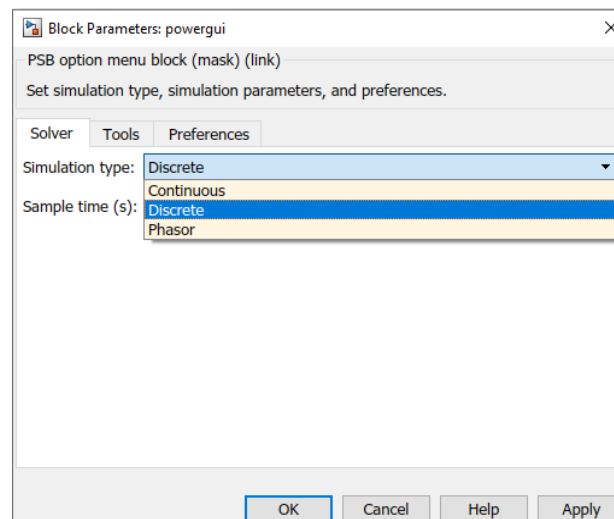


Figura 4-4: Configuración de *solver* powergui Simulink

Para llevar a cabo este tipo de simulaciones hay que incorporar un bloque *powergui* al esquemático donde se detalla el tipo de simulación (discreta o continua) como se puede observar en la Figura 4-4, el tipo de *solver* y otra serie de parámetros para personalizar el comportamiento del circuito. Para la simulación teórica se escogió el tipo de simulación continua, ya que es el que permite desactivar completamente todas las no idealidades y

pérdidas que pueda introducir cada componente, especialmente para el MOSFET y el diodo. Las características activadas aparecen en la Figura 4-5.

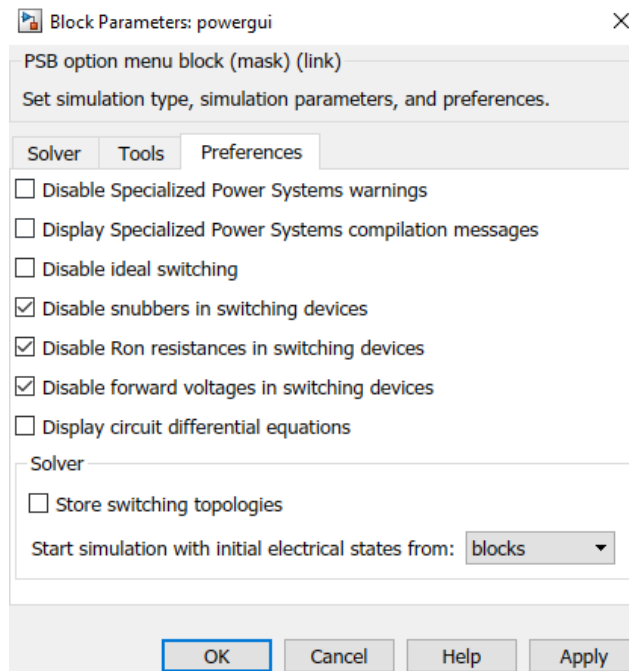


Figura 4-5: Parámetros simulación ideal Simulink

Las *waveforms* obtenidas para ambas medidas a comparar (I_L y V_O) aparecen representadas en las Figuras 4-6 y 4-7.

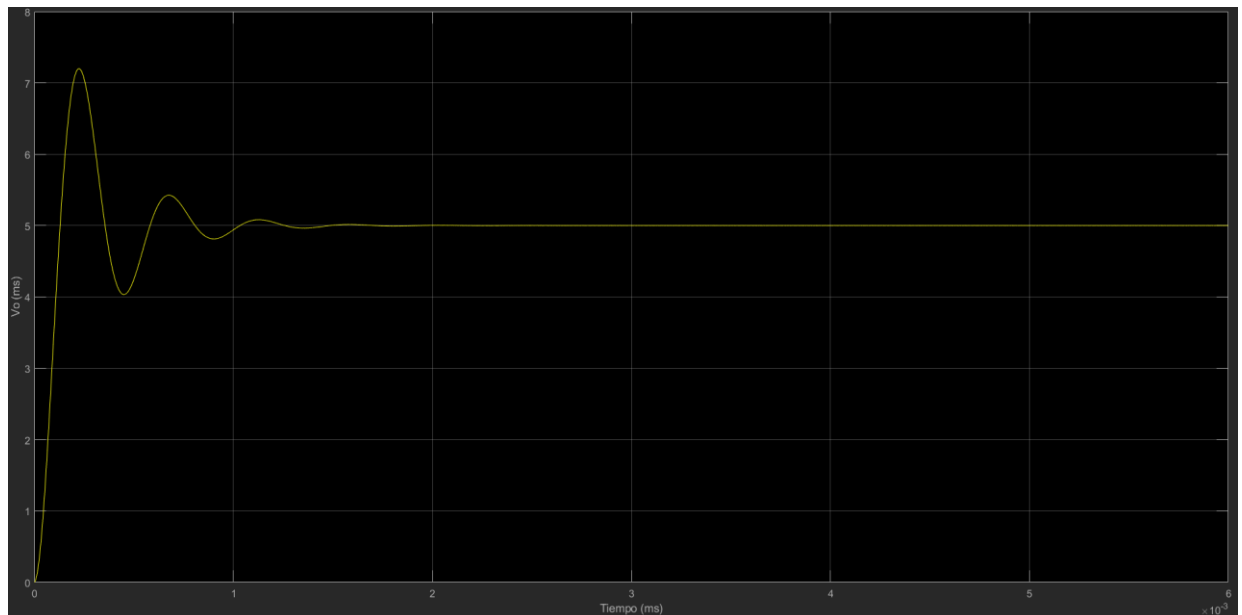


Figura 4-6: Medida V_O teórica Simulink (sin pérdidas)

Como se puede observar la tensión de salida alcanza en régimen permanente el valor deseado de 5 V. Además, se aprecia claramente una forma típica de un segundo orden, con una sobreoscilación inicial que llega hasta los 7,2 V y con un tiempo de establecimiento, con el criterio al 5 %, de unos 0,74 ms.

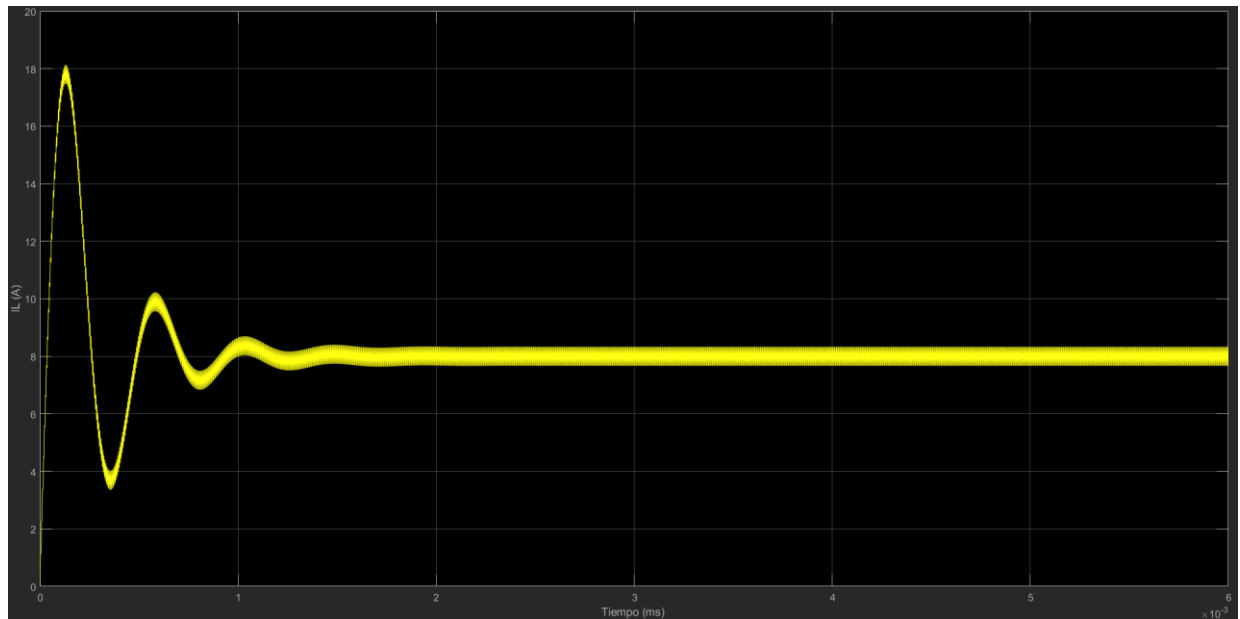


Figura 4-7: Medida I_L teórica Simulink (sin pérdidas)

En cuanto a la corriente por la bobina se aprecia una forma similar, con una sobreoscilación inicial de hasta los 18 A y se consigue en régimen permanente una corriente media de 8 A con un rizado de 0,66 A.

4.2.2 Typhoon HIL

En este apartado se procederá a montar y simular el convertidor *buck* diseñado sobre la herramienta Typhoon HIL, sin la incorporación de pérdidas.

Lo primero que se debe hacer tras ejecutar el *software* Typhoon HIL Control Center, tal y como se vio en el apartado 3.1.2, es acceder al editor de esquemáticos o Schematic Editor. Una vez dentro, se procede a crear un nuevo esquemático al que se debe dar un nombre y, a continuación, se comienza a realizar el diseño arrastrando los diferentes bloques y componentes, necesarios para lograr la funcionalidad deseada, de las bibliotecas al esquemático vacío. La Figura 4-8 muestra el conjunto de componentes de electrónica de potencia ofrecida por la herramienta, destacando la variedad de elementos convertidores ofrecidos.

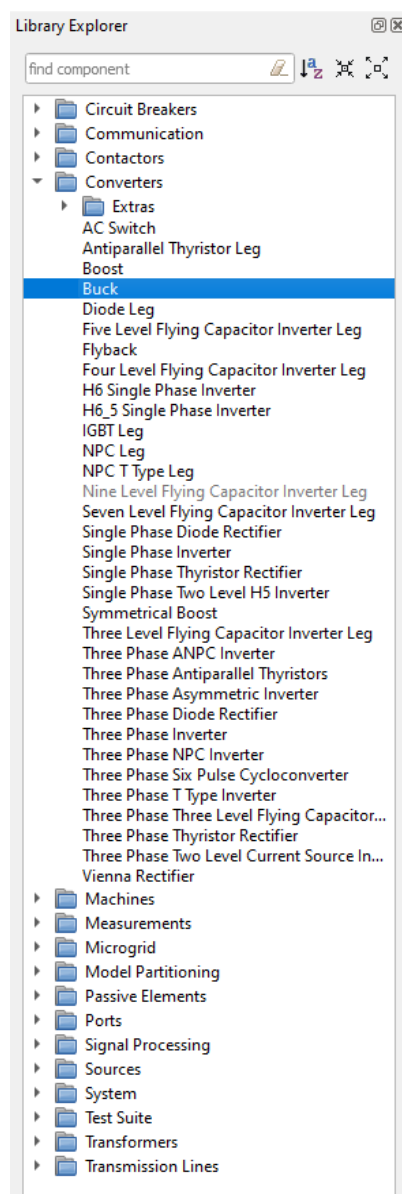


Figura 4-8: Bibliotecas de componentes y bloques Typhoon HIL

Cabe destacar que Typhoon HIL es bastante específico en cuanto al tipo de componentes que se pueden usar, puesto que hay que usar bloques de alto nivel ya implementados por ellos como un *buck* entero. No permite introducir un sencillo transistor o un diodo para diseñar los convertidores manualmente, por ejemplo. Y esto como se verá en el apartado 4.4, limita las pérdidas que se podrán introducir finalmente en el circuito.

La Figura 4-9 muestra el circuito *buck* completo, una vez incluidos los diferentes bloques y componentes.

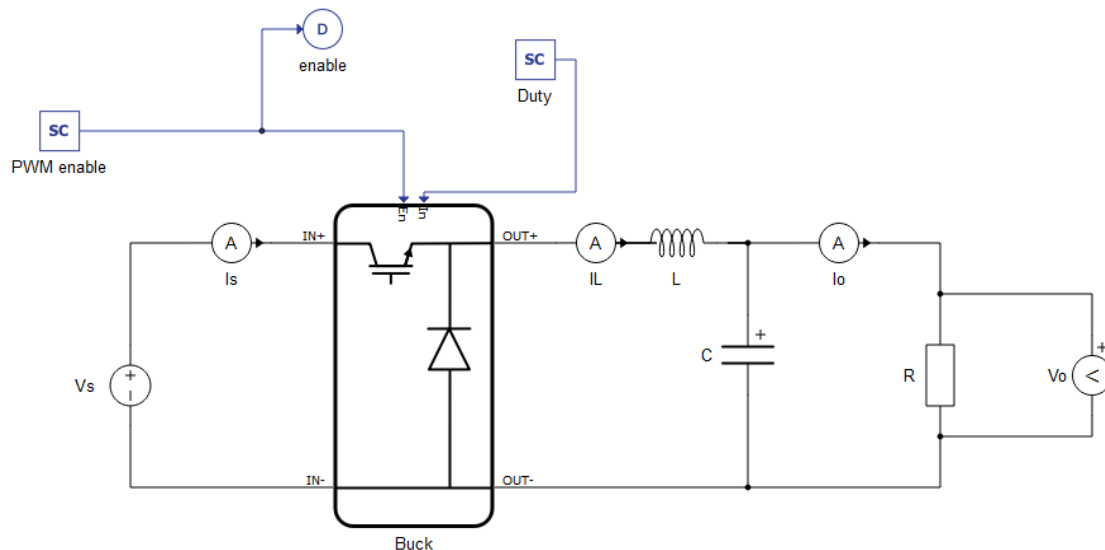


Figura 4-9: Circuito *buck* Typhoon HIL (sin pérdidas)

La bobina, el condensador y la carga llevan los valores mostrados en el apartado 4.1. Además, se incluyen amperímetros para medir las corrientes de entrada y salida y la corriente por la bobina, así como un voltímetro para medir el voltaje de salida en la carga.

En cuanto al bloque *buck*, en la Figura 4-11 se muestran las características seleccionadas para la simulación. Dicho bloque tiene dos modos posibles de control: o por modulador interno (*Internal Modulator*) o por entradas digitales (*Digital Inputs*). En este caso se eligió la opción de modulador interno ya que permite más posibilidades de personalización y la selección de la frecuencia de conmutación de forma sencilla. Dicha frecuencia es la diseñada de 200 kHz. El periodo de tiempo muerto o *Dead time period* es importante dejarlo a cero, ya que se está trabajando con un *buck* asíncrono. Por último, para el tema de la referencia de la señal hay varias posibilidades igualmente válidas para que funcione con el ciclo de trabajo diseñado. La opción más sencilla e intuitiva, y la que se ha empleado finalmente consiste en referenciar la señal entre 0 y 1, para que después al introducir el *Duty* a la entrada *In* del bloque *buck*, simplemente haya que introducir su valor de 41,667%. Limitando así en valor del ciclo de trabajo entre 0 y 1 (0 y 100 %). Pero existe otra posibilidad, ya que el bloque por defecto viene referenciado entre -1 y 1. En este caso la entrada al *buck* no podría ser directamente 0,41667, ya que el programa lo entiende como un *Duty* de 1,41667 sobre un rango de dos unidades (entre -1 y 1). Por ello sería en realidad un *Duty* de $1,41667/2$, es decir del 70,83 % y, en lugar de obtener a la salida 5 V con una entrada de 12 V se conseguiría una mayor, de 8,5 V. Luego, en este caso, para obtener 5 V de salida, habría que introducir un “*Duty* negativo” a ojos del programa de:

$$\frac{1 + Duty}{2} = 0,41667$$

$$Duty = -0,1667$$

En la Figura 4-10 se ilustran varias posibilidades para obtener un *Duty* de 41,667% a ojos del programa, teniendo en cuenta las distintas referencias.

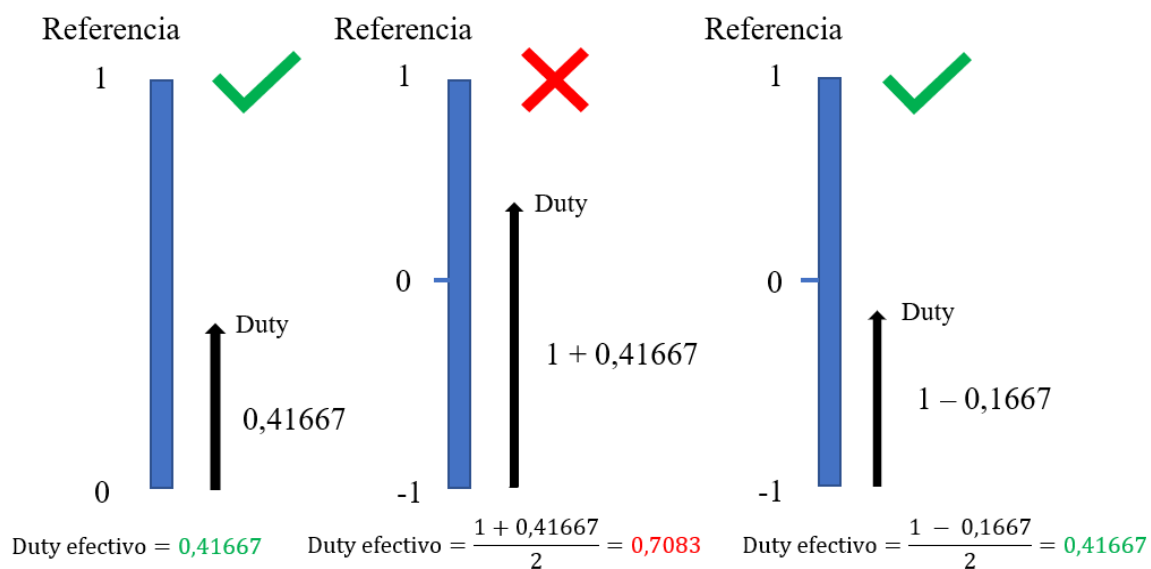


Figura 4-10: Obtención *Duty* 0,41667 con varias referencias Typhoon HIL

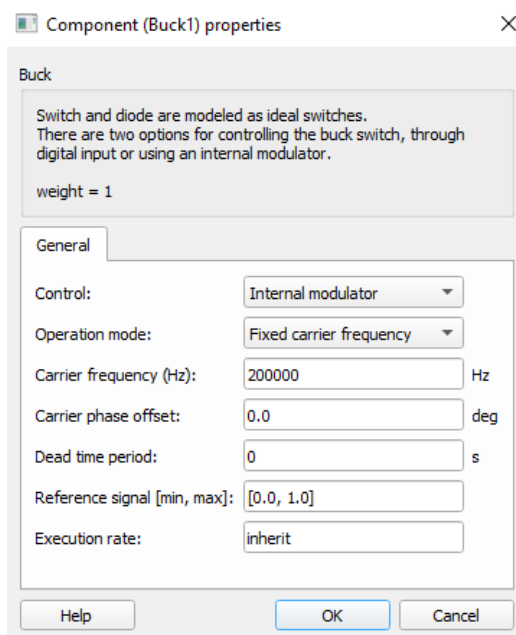


Figura 4-11: Configuración bloque *buck* Typhoon HIL

Las entradas del *buck* son un *Enable* que se mantendrá siempre activo y el ciclo de trabajo. Otro paso importante previo a la simulación del circuito consiste en la configuración de los parámetros del esquemático o *Schematic Settings*, como se empezó a mostrar en el apartado 3.1.2. Para ello, se selecciona el *hardware* HIL402 sobre el que se realizará la simulación y su configuración, así como el tipo de *solver* (euler, exact, trapezoidal) y el paso de simulación o “dt”.

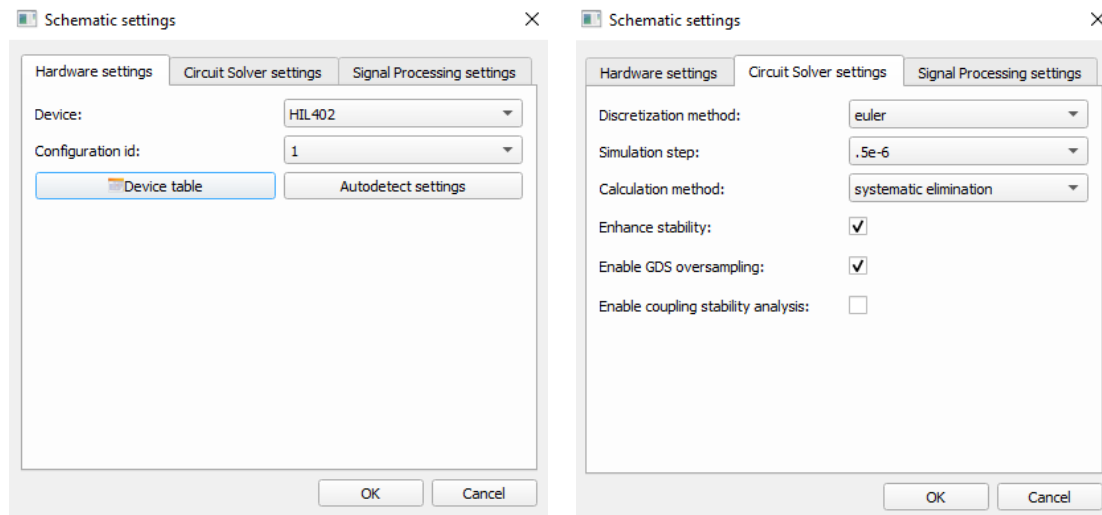


Figura 4-12: Parámetros esquemático Typhoon HIL

Como tipo de *solver* se seleccionará Euler, ya que es el que también se empleará en el diseño en LabVIEW, de modo que se pueda establecer una comparativa adecuada. Para el paso de simulación se elegirá 0,5 μ s (500 ns), ya que es el paso más pequeño permitido por la herramienta.

Una vez configurado todo el modelo se procede a compilar y, una vez que este proceso se completa correctamente se pueden observar el reporte con los resultados en cuanto a utilización. En las Tabla 4-1, 4-2 y 4-3 se ha recopilado de forma ordenada dicho reporte y se ha ordenado en tres partes: utilización del *hardware* específico, utilización por *core* y utilización de *software*.

Utilización <i>hardware</i> específico			
Descripción	Utilizado	Disponible	Porcentaje
Standard processing core	1	3	33,33 %
Signal generator	1	12	33,33 %
Look up tables	0	8	0 %
Machine solver	0	1	0 %
Parallel DTV Conv. Detectors	0	0	0 %
PWM channels	1	12	8,33 %

Tabla 4-1: Utilización HW específico Typhoon HIL

Utilización <i>core0</i>			
Descripción	Utilizado	Disponible	Porcentaje
Power Electronics Converters	1	3	33,33 %
Contactor	0	6	0 %
TVE solvers	0	0	0 %
SP sources	0	16	0 %
Delayed controlled sources	0	12	0 %
Non-ideal switches	0	0	0 %
Matrix memory	-	-	0,88 %
Time slot of core	-	-	60%

Tabla 4-2: Utilización *core0* Typhoon HIL

Utilización <i>software</i>			
Descripción	Utilizado	Disponible	Porcentaje
Signal processing IO variables	2	4032	0,049 %
Signal processing Probes	0	512	0 %
Signal processing Digital Probes	1	128	0,78 %
Total utilization of the internal memory (kB)	46	255	18,04 %
Code segment size (kB)	33	255	12,94 %
Data segment size (kB)	13	255	5,098 %

Tabla 4-3: Utilización *software* Typhoon HIL

Nótese que se ha empleado la configuración 1 del *hardware*, la cual cuenta con tres *cores*, de los que únicamente se usa uno. Otras configuraciones como la 2 permiten hasta un máximo de 4 *cores*. Como se puede apreciar en el reporte el *core* tiene hasta tres convertidores de potencia y es destacable al analizar los resultados que Typhoon HIL, a diferencia de otras soluciones comerciales, no muestra la utilización a bajo nivel de componentes ya que, aunque aparecen por ejemplo LUTs, éstas no se emplean y lo que sí se usan son los bloques convertidores ya integrados por ellos.

A continuación, se abre el panel HIL SCADA donde se realiza la simulación HIL a tiempo real. En esta parte es posible incorporar distintos *widgets* y demás indicadores para poder visualizar y modificar las diferentes señales. En la simulación realizada se usaron los componentes que aparecen en la Figura 4-13.

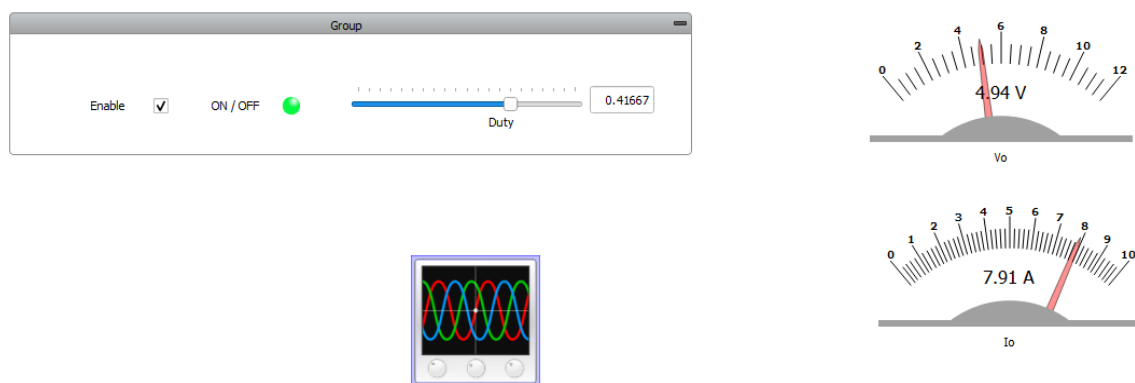


Figura 4-13: Widgets HIL SCADA simulación *buck*

- Dos “*Gauges*” para mostrar a tiempo real el valor del voltaje de salida (V_O) y la corriente por la bobina (I_L).
- Un grupo de *widgets* formado por los siguientes componentes:
 - Una “*Checkbox*” para seleccionar si se desea habilitar el *buck*.
 - Un “*LED*” que indica si el *enable* se encuentra activo.
 - Un “*Slider*” para variar a tiempo real el valor del *Duty* (entre 0 y 1).
- Un visor “*Capture/Scope*” que nos permite observar las distintas señales medidas a modo osciloscopio, o bien a tiempo real, o bien se pueden capturar para analizar y exportar. Este será el componente más importante para visualizar la forma y valor de las diferentes señales.

Además, se deben inicializar las diferentes señales de entradas como el voltaje de entrada a 12 V y el *Duty* a su valor de 5/12. Esto se puede hacer desde el panel de “*Model Settings*” y aparece representado en la Figura 4-14.

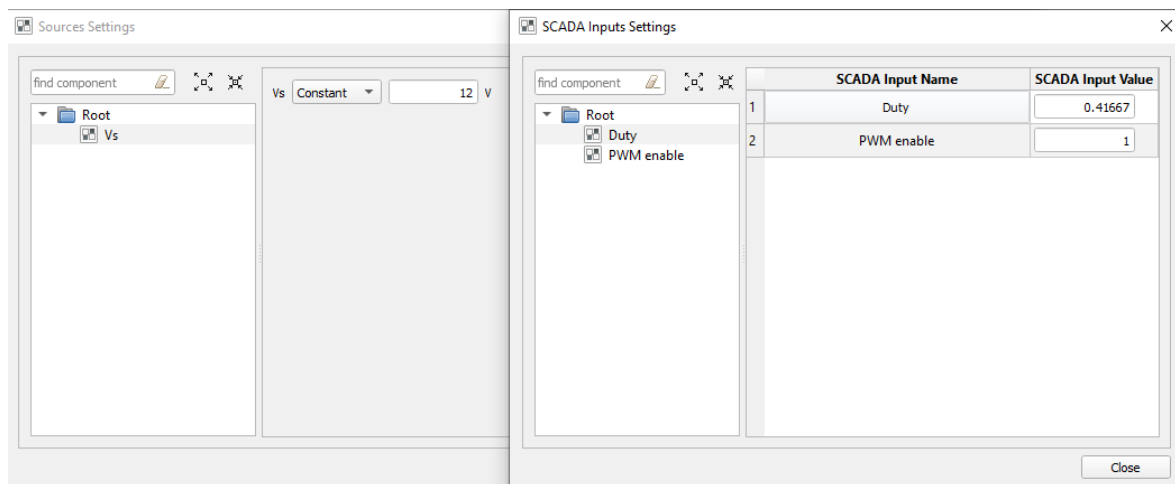


Figura 4-14: Inicialización parámetros de entrada HIL SCADA

En la Figura 4-15 aparece la simulación a tiempo real de varias señales tal y como la saca el objeto visor en modo “*Scope*”. Este tipo de visualización aporta sobre todo información acerca del régimen permanente de las señales y cómo éste va variando al tocar diferentes parámetros.

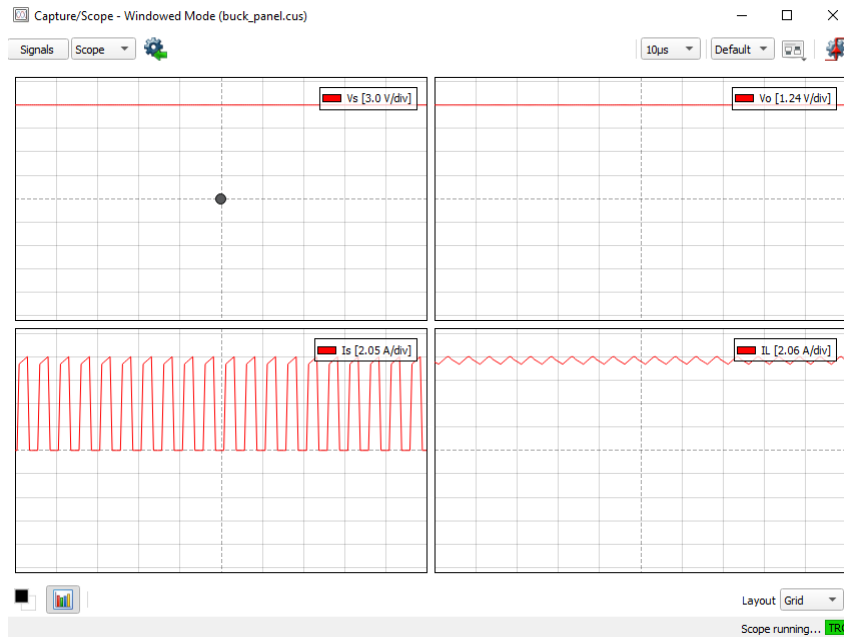


Figura 4-15: Simulación Scope *buck* HIL SCADA

Para visualizar los instantes iniciales, es necesario configurar el *trigger* y capturar las señales mediante la opción “*Capture*” de este mismo componente. En la Figura 4-16 aparecen capturadas las señales: V_s , V_o , I_s e I_L , durante 10 ms.

Ésta será la simulación que se exportará y se usará para realizar la comparativa frente a otras herramientas comerciales. En concreto, se hará sobre las señales de la corriente por la bobina (I_L) y el voltaje de salida (V_o).



Figura 4-16: Simulación Capture *buck* HIL SCADA (sin pérdidas)

También es posible añadir cursores para observar los detalles de las señales en diferentes puntos, como se ejemplifica en la Figura 4-17, sobre un punto temporal durante el periodo de régimen permanente.









Signal Name	Cursor 1
▼ Viewport 1	
 Time	0.005s
 Vs	12.0
▼ Viewport 2	
 Time	0.005s
 Vo	4.9434414
▼ Viewport 3	
 Time	0.005s
 Is	7.730234
▼ Viewport 4	
 Time	0.005s
 IL	7.730234

Figura 4-17: Cursor simulación Capture *buck* HIL SCADA (sin pérdidas)

Cabe destacar su gran parecido al caso ideal realizado en Simulink, tanto en forma como en valores de voltaje y corriente, máxima sobreoscilación y en régimen permanente, como en tiempo de establecimiento.

- Puesta al límite de la herramienta:

Como se ha observado, la utilización de los componentes *hardware* al montar el *buck*, dista de ser la máxima permitida por las diferentes configuraciones del dispositivo HIL402. Para forzar esta situación, se ha montado un escenario, empleando la configuración 2, que como muestra la Figura 3-16 cuenta con el mayor número de núcleos, es decir, 4. Se ha probado simular 4 *bucks* idénticos como se ilustra en la Figura 4-18.

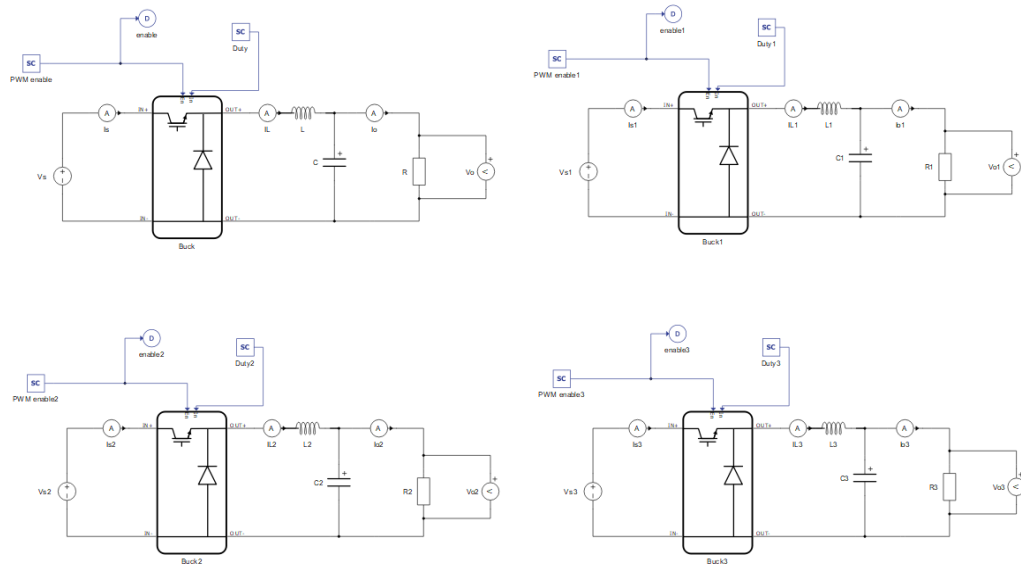


Figura 4-18: Esquemático 4 *bucks* Typhoon HIL

A continuación, se compila el modelo para ver si el programa arroja algún error que, como se muestra en la Figura 4-19, no es así. Se observa cómo se está usando el 100 % de los *cores* (los 4 *cores*) y, dentro de cada uno, se permite hasta el uso de hasta 3 convertidores. Por ello, se puede afirmar que es posible montar una gran cantidad de convertidores conmutados, sin llegar a la capacidad máxima de la herramienta.

```
Running Device specific hw utilization analysis:
Standard processing core utilization: 4 out of 4 100.0%
Signal generator utilization: 4 out of 12 33.33%
Look up tables utilization: 0 out of 8 0.0%
Machine solver utilization: 0 out of 0 0.0%
Parallel DTV Conv. Detectors utilization: 0 out of 0 0.0%
PWM channels utilization: 4 out of 12 33.33%

Running core0 specific hardware utilization analysis:
Power Electronics Converters utilization: 1 out of 3 33.33%
Contactor utilization: 0 out of 6 0.0%
TVE solvers utilization: 0 out of 0 0.0%
SP sources utilization: 0 out of 16 0.0%
Delayed controlled sources utilization: 0 out of 12 0.0%
Non-ideal switches utilization: 0 out of 0 0.0%

Running core1 specific hardware utilization analysis:
Power Electronics Converters utilization: 1 out of 3 33.33%
Contactor utilization: 0 out of 6 0.0%
TVE solvers utilization: 0 out of 0 0.0%
SP sources utilization: 0 out of 16 0.0%
Delayed controlled sources utilization: 0 out of 12 0.0%
Non-ideal switches utilization: 0 out of 0 0.0%

Running core2 specific hardware utilization analysis:
Power Electronics Converters utilization: 1 out of 3 33.33%
Contactor utilization: 0 out of 6 0.0%
TVE solvers utilization: 0 out of 0 0.0%
SP sources utilization: 0 out of 16 0.0%
Delayed controlled sources utilization: 0 out of 12 0.0%
Non-ideal switches utilization: 0 out of 0 0.0%

Running core3 specific hardware utilization analysis:
Power Electronics Converters utilization: 1 out of 3 33.33%
Contactor utilization: 0 out of 6 0.0%
TVE solvers utilization: 0 out of 0 0.0%
SP sources utilization: 0 out of 16 0.0%
Delayed controlled sources utilization: 0 out of 12 0.0%
Non-ideal switches utilization: 0 out of 0 0.0%
```

Figura 4-19: Utilización 4 *bucks* Typhoon HIL

A la hora de simularlo se seleccionan cuatro ciclos de trabajo diferentes, para sacar a la salida, ante 12 V de entrada: 3 V, 5 V, 6 V y 9,6 V, como se enseña en la Figura 4-20.

	SCADA Input Name	SCADA Input Value
1	Duty	0.25
2	Duty1	0.41667
3	Duty2	0.5
4	Duty3	0.8

Figura 4-20: 4 Ciclos de trabajo *bucks*

Se capturan las señales de tensión de salida como se muestra en la Figura 4-21 y se inserta en el régimen permanente de todas ellas un cursor para observar que los resultados de tensión son los adecuados para cada uno de los *bucks*, como se aprecia en la Figura 4-22.

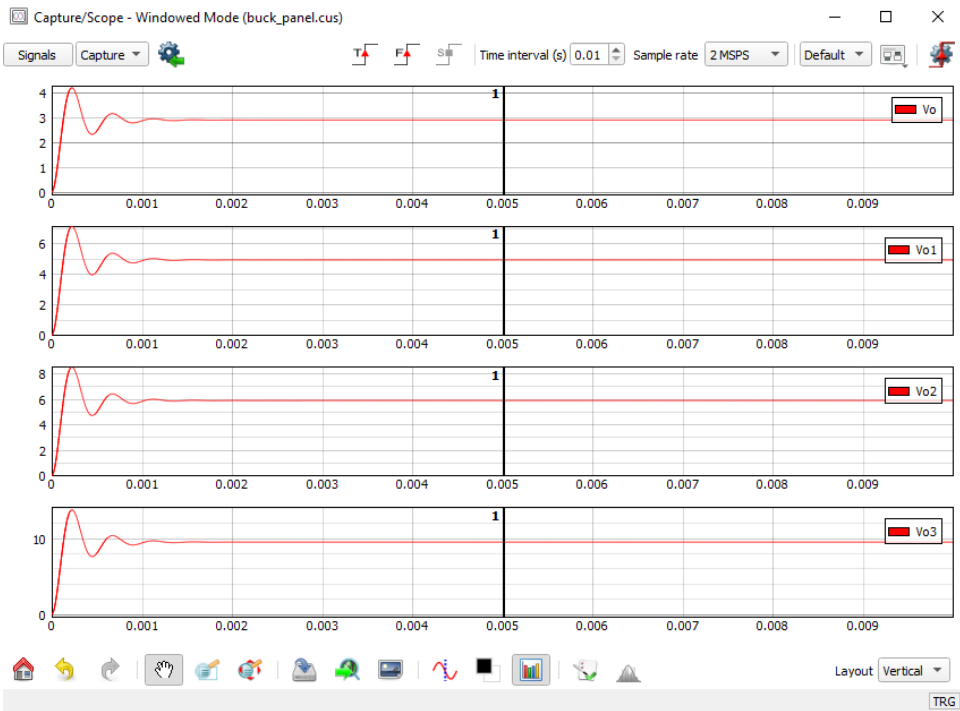


Figura 4-21: Simulación Capture 4 *bucks* Typhoon HIL

Signal Name	Cursor 1
▼ Viewport 1	
⌚ Time	0.005s
Vo	2.9278111
▼ Viewport 2	
⌚ Time	0.005s
Vo1	4.9434414
▼ Viewport 3	
⌚ Time	0.005s
Vo2	5.9033046
▼ Viewport 4	
⌚ Time	0.005s
Vo3	9.551653

Figura 4-22: 4 Cursores simulación Capture *buck* Typhoon HIL

4.2.3 LabVIEW Control Design and Simulation Module + Multisim

Para un primer acercamiento de despliegue con herramientas de National Instruments, se estudiará lo comentado en el apartado 3.2.2 respecto al montaje sobre la plataforma LabVIEW con el *toolkit* “Control Design and Simulation Module” y el programa Multisim, a través del ejemplo concreto del *buck* definido. El aspecto del circuito diseñado en Multisim se muestra en la Figura 4-23, y será explicado con detalle.

Aunque como bien se ve, el circuito del *buck* se ha implementado mediante diodo e interruptor, es cierto que existe un bloque *buck* ya diseñado que se puede emplear obteniendo la misma funcionalidad, pero al exportarlo a LabVIEW aparecían errores relacionados con dicho bloque, por lo que se optó por este método.

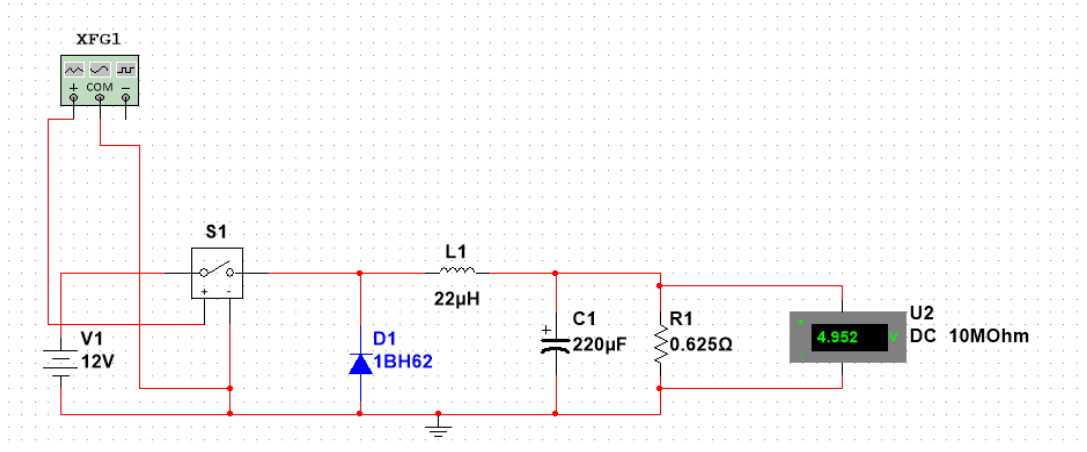


Figura 4-23: Simulación circuito *buck* Multisim

A la salida se ha situado un voltímetro para medir el voltaje de salida en la carga. Cabe destacar que es necesario generar la señal PWM para hacer el *switching* en el interruptor a través de un generador de funciones, en modo señal cuadrada, introduciendo la frecuencia de conmutación, el ciclo de trabajo, la amplitud y el *offset*. En la Figura 4-24 se muestra la generación de la señal PWM, como aclaración, nótese que la amplitud se encuentra a 5 V dado que el interruptor se encuentra referenciado a 5 V.

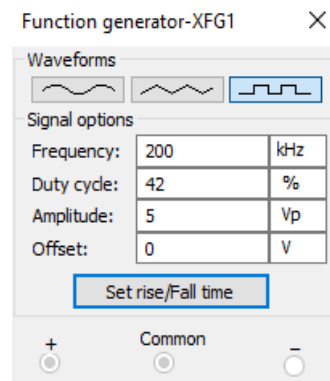


Figura 4-24: Generación de señal PWM circuito *buck* Multisim

Tras verificar que los valores de tensión de salida son los correctos y esperados, se procede a acondicionar el circuito para su exportación a la plataforma LabVIEW. Para ello es necesario sustituir las señales de entrada y salida por una serie de pines en la dirección adecuada. La Figura 4-25 ilustra la adecuación del circuito para su funcionamiento a modo de “caja negra” en LabVIEW. Como se observa, las entradas que admitirá el bloque son dos: tensión de entrada y ciclo de trabajo o *Duty*, y la única salida será el voltaje de salida. El control de la tensión a la entrada se lleva a cabo a través de una fuente de voltaje regulada por voltaje, de modo que cuando en LabVIEW se introduzca una tensión determinada al bloque, ésta se traduzca al mismo voltaje de entrada de manera interna al circuito de Multisim.

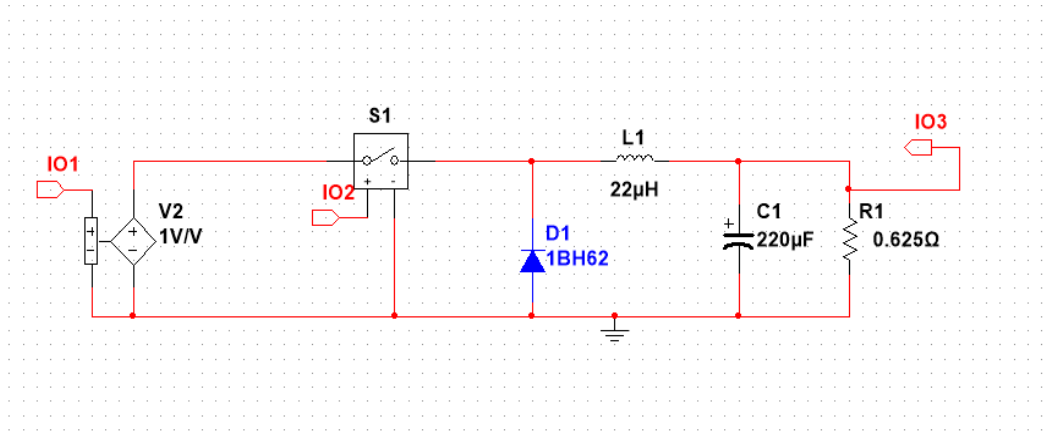


Figura 4-25: Circuito Multisim a exportar a LabVIEW

En LabVIEW se parte del módulo “Control Design and Simulation”, que está especialmente diseñado para observar el comportamiento en lazo abierto de sistemas, implementar controladores en lazo cerrado, simular sistemas dinámicos e implementar modelos y sistemas de control HW a tiempo real. Cuenta con numerosos bloques propios con los que se pueden implementar controladores PID, generar diferentes señales, gráficos para visualizar resultados, etc. Además, es especialmente interesante el tema de que dentro de su tipo de bucle principal (el “Control & Simulation Loop”) se permite la inclusión de modelos externos de varios programas *software*, entre los que se encuentra Multisim.

En la Figura 4-26 se muestra la incorporación del modelo *buck*, diseñado en la plataforma Multisim, en el diagrama de bloques de un proyecto en LabVIEW. Como se puede observar, se conecta la tensión de entrada a un control para alterar su valor en tiempo real según se desarrolle la simulación. El *Duty* se conecta a un bloque generador de señal cuadrada, propio de este módulo que hará las funciones de conmutación PWM. Por último, tanto la tensión de entrada como la de salida se agrupan en un *array* y se llevan juntas a una gráfica para su posterior visualización.

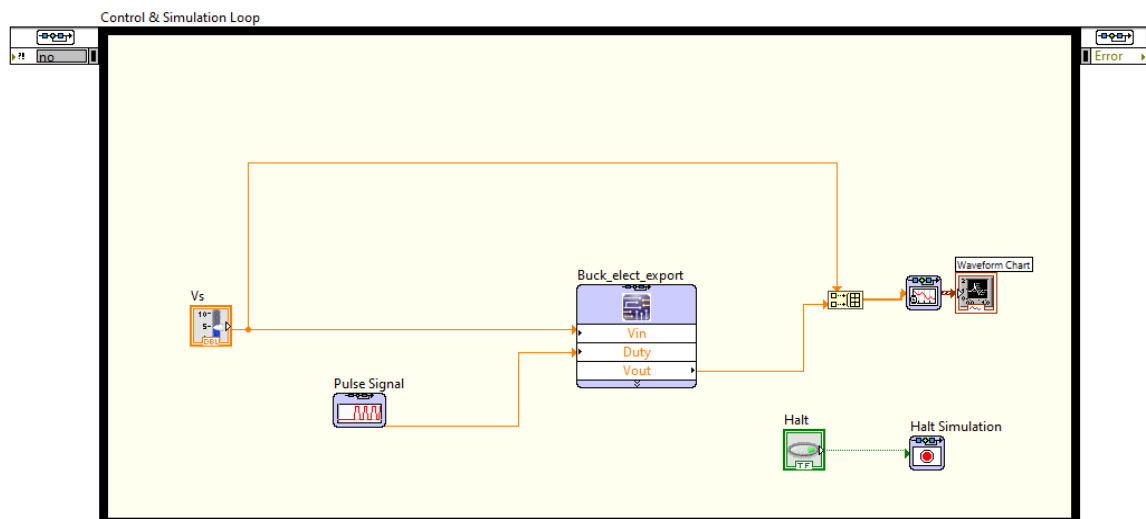


Figura 4-26: Circuito *buck* LabVIEW Control Design and Simulation + Multisim

Los parámetros del bloque generador de PWM se muestra en la Figura 4-27. Se trata básicamente de la misma funcionalidad del generador de señales que había en Multisim (Figura 4-24) y en él se detalla: el tiempo de inicio de la señal, amplitud, *offset*, periodo y ciclo de trabajo.

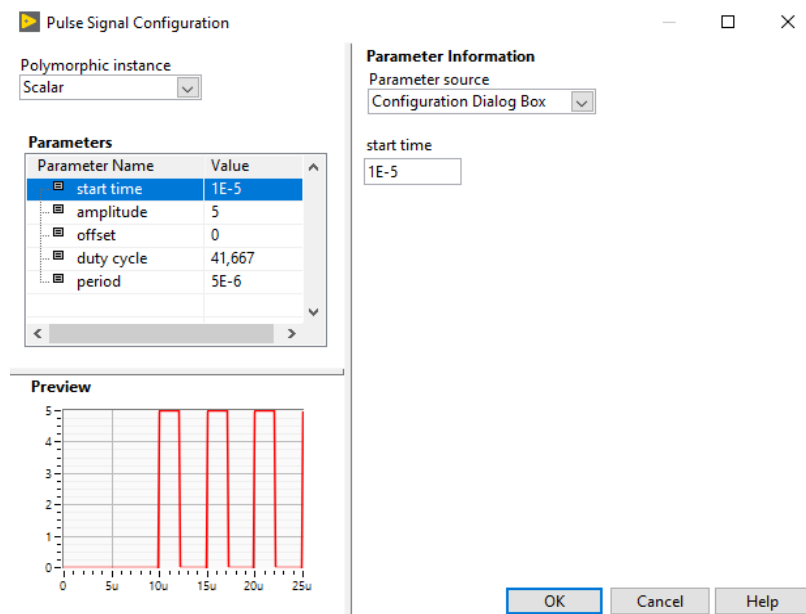


Figura 4-27: Generación de señal PWM circuito *buck* LabVIEW

En la Figura 4-28 se muestran las opciones en cuanto a parámetros de las que dispone el bucle principal “Control & Simulation Loop” propio de este módulo de LabVIEW.

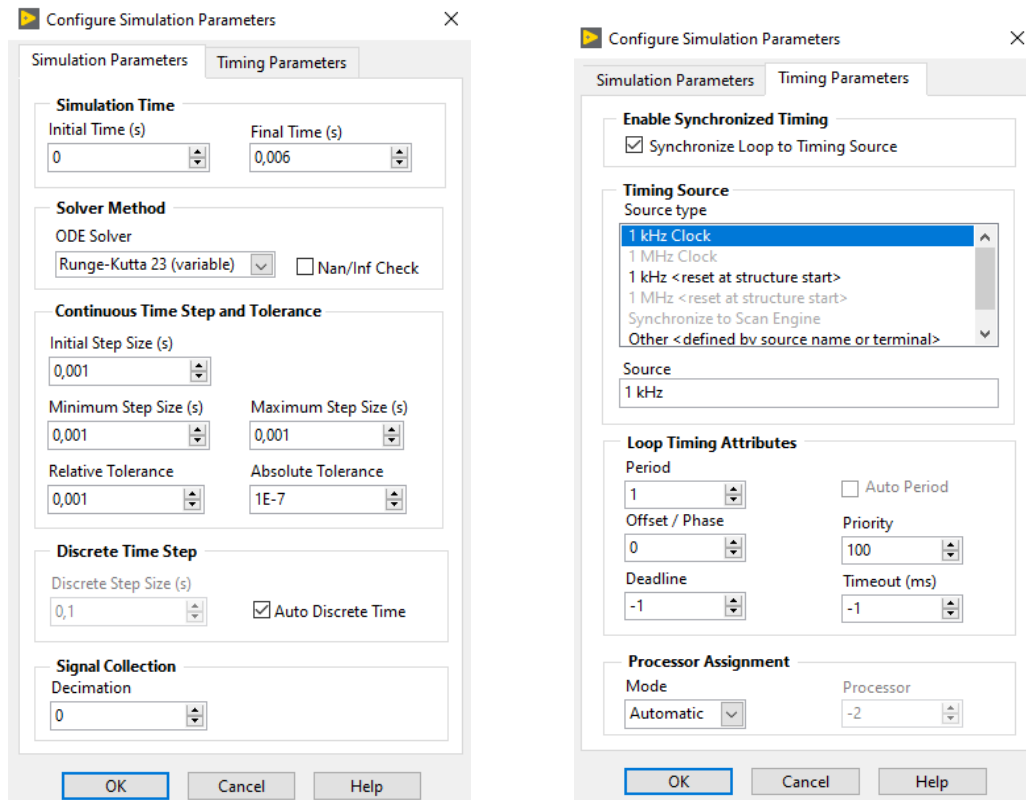


Figura 4-28: Parámetros bucle Control & Simulation

Este tipo de simulación admite una gran variedad de *solvers*, como Runge-Kutta o Euler, además se permite sincronizar el bucle a una determinada frecuencia de reloj que, como máximo será de 1 kHz, pudiendo superarse (1 MHz o más) al conectar algún *hardware* específico apropiado, en cuyo caso la simulación no será a tiempo real, ya que como máximo se permite 1 kHz en RT. Si se sincroniza el paso de simulación a dicha frecuencia máxima, es posible llevar a cabo una simulación a tiempo real del sistema, obteniendo la gráfica que se muestra en la Figura 4-29 que, como se ha observado en el apartado 4.2.1, se trata de resultados correctos, tanto en tiempo como en valores de voltaje de salida.

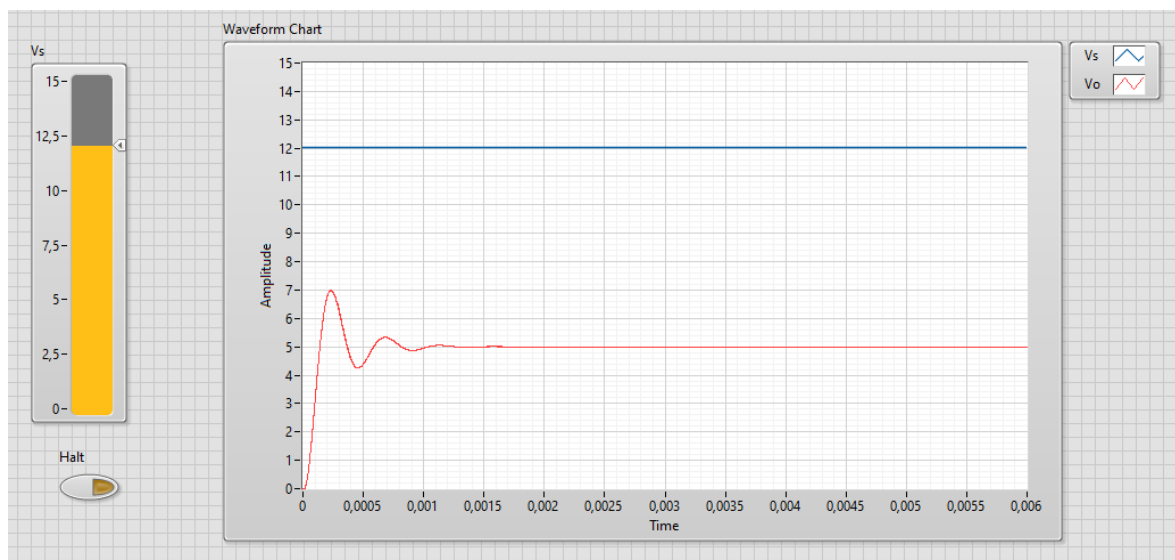


Figura 4-29: Resultado Voltaje de salida simulación con Multisim

El problema que presenta este acercamiento frente a lo que se pretende alcanzar en este trabajo, consiste en primer lugar en que se trata ante todo en una simulación *software* que, si bien puede ser muy útil para desplegar sistemas y visualizar resultados de manera rápida, no explota en demasía las características específicas de un *hardware* concreto. Además, en este tipo de simulaciones no es posible visualizar la utilización de recursos al desplegar el modelo a tiempo real en una máquina determinada, algo que se busca comparar en este proyecto. Por último, cabe destacar que la máxima frecuencia permitida para simular a tiempo real de 1 kHz se queda bastante corta si se compara con el rango típico de frecuencias sobre el que se suele trabajar en este tipo de convertidores conmutados.

De modo que, de cara a poder aprovechar al máximo la capacidad de un *hardware* concreto, en el caso de este trabajo, en NI myRIO-1900 descrito en 3.2.1, y poder visualizar la utilización concreta de un despliegue sobre dicha plataforma, es necesario diseñar a un nivel inferior: a nivel de FPGA. Este tipo de diseño, que se muestra en los apartados 4.2.4 y 4.3.3, permite configurar con mucha mayor profundidad y precisión todos los parámetros y características del sistema, pero, como contrapartida, el modelado en sí es mucho menos intuitivo y cuenta con un grado de complejidad para nada despreciable ante un usuario sin experiencia en el campo en el que se enmarca este trabajo.

4.2.4 LabVIEW FPGA

En este apartado se procederá a la exposición del montaje y la simulación realizados para lograr el comportamiento del convertidor *buck* especificado en el apartado 4.1 a través de la herramienta LabVIEW FPGA.

Como se ha mencionado con anterioridad, LabVIEW FPGA no permite un diseño sencillo modo esquemático de tipo SPICE, por lo que se ha de partir de las ecuaciones que modelan el circuito específico para incluirlas en los componentes de bajo nivel, estructuras y bucles de LabVIEW y así lograr simular el comportamiento circuital deseado. Este hecho es en cierto modo un inconveniente que presenta esta herramienta ya que, como se va a ver a continuación, no es para nada trivial el diseño de un circuito de complejidad media como un convertidor conmutado con esta metodología.

Para llevar a cabo esta tarea hay varias aproximaciones, como por ejemplo la detallada en [21]. En este trabajo se consideró partir como base del modelo realizado sobre esta herramienta en un artículo de *Energies* [6], y luego realizar las modificaciones necesarias sobre éste para el convertidor específico que se propone en este trabajo.

Ya se han visto los circuitos para cada uno de los tres escenarios de funcionamiento posibles del circuito *buck*, en el apartado 2.3 (Figuras 2-10, 11 y 12), y las ecuaciones en diferencias que modelan su comportamiento. En la siguiente tabla se muestra a modo de resumen la solución para cada parámetro definido en función del escenario analizado.

	SWITCH “ON” & DIODO “ABIERTO”	SWITCH “OFF” & DIODO CONDUCE	SWITCH “OFF” & DIODO “ABIERTO”
$\frac{\partial i_L}{\partial t}$	$\frac{V_S - i_L(R_L + R_{DS_{ON}}) - V_O}{L}$	$\frac{-i_L R_L - V_O - V_D}{L}$	0
$\frac{\partial v_C}{\partial t}$	$\frac{1}{C} \left(i_L - \frac{V_O}{R} \right)$	$\frac{1}{C} \left(i_L - \frac{V_O}{R} \right)$	$-\frac{V_O}{C R}$
V_O	$\frac{R}{R + ESR} (i_L ESR + V_C)$	$\frac{R}{R + ESR} (i_L ESR + V_C)$	$\frac{V_C R}{R + ESR}$

Tabla 4-4: Ecuaciones en diferencias modelo *buck* asíncrono

Una vez que se tienen las ecuaciones diferenciales a resolver, es necesario seleccionar un método matemático para resolverlas. En el modelo propuesto en el artículo se opta por el método de Euler dada la simplicidad de su algoritmo, lo que supone un menor tamaño *hardware* y, por tanto, resulta en frecuencias máximas mayores o, equivalentemente, pasos temporales menores. Como se ha visto en el apartado 4.2.2, en Typhoon está disponible la opción de simular mediante un *solver* tipo Euler. Por ello y, para que la comparativa sea equivalente y tenga sentido se usará el método Euler en ambas herramientas.

En la Tabla 4-5 se muestran las soluciones de las ecuaciones anteriores resueltas mediante el método de Euler para las tres incógnitas presentadas anteriormente: i_L o corriente por la bobina, V_c o tensión en el condensador y V_o o voltaje de salida en la carga. Los subíndices indican el instante de cada solución, de modo que aquellos parámetros con subíndice k se corresponden con el valor actual, mientras que los que tienen el subíndice $k+1$ son los valores siguientes de la solución.

	SWITCH “ON” & DIODO “ABIERTO”	SWITCH “OFF” & DIODO CONDUCE
$i_L(k+1)$	$i_L(k) + \frac{h}{L} (V_s(k) - i_L(k)(R_L + R_{DS_{ON}}) - V_o(k))$	$i_L(k) - \frac{h}{L} (i_L(k) R_L + V_o(k) + V_D(k))$
$V_c(k+1)$	$V_c(k) + \frac{h}{C} \left(i_L(k) - \frac{V_o(k)}{R} \right)$	$V_c(k) + \frac{h}{C} \left(i_L(k) - \frac{V_o(k)}{R} \right)$
$V_o(k)$	$\frac{R}{R + ESR} (i_L(k) ESR + V_c(k))$	$\frac{R}{R + ESR} (i_L(k) ESR + V_c(k))$

	SWITCH “OFF” & DIODO “ABIERTO”
$i_L(k+1)$	0
$V_c(k+1)$	$V_c(k) - \frac{h}{C} \left(\frac{V_o(k)}{R} \right)$
$V_o(k)$	$\frac{V_c(k) R}{R + ESR}$

Tabla 4-5: Soluciones ecuaciones en diferencias método Euler modelo *buck* asíncrono

Los diferentes parámetros y variables de la tabla anterior se resumen a continuación:

- i_L : corriente por la bobina
- V_c : voltaje en el condensador
- V_o : voltaje en la carga
- V_s : voltaje de entrada
- V_D : caída de voltaje del diodo
- R : resistencia de la carga
- R_L : resistencia serie de la bobina
- $R_{DS_{on}}$: resistencia del interruptor en estado ON
- ESR : resistencia serie equivalente del condensador

A los términos denominados h/L y h/C en el modelo, debe dárseles un valor de modo que sea constante durante la simulación. El valor tanto de la bobina como del condensador a emplear ya se conocen, solo quedaría ajustar el paso temporal o *time step*, es decir, la h , que no es otra cosa que el paso de simulación o “ dt ” escogido, luego los términos h/L y h/C serían en realidad dt/L y dt/C si se sigue la nomenclatura de este trabajo. Como se ha visto en el apartado 4.2.2, Typhoon HIL tiene como límite mínimo de paso de simulación 500 ns, por ello, se usará ese mismo valor en LabVIEW de modo que la comparativa entre ambas simulaciones tenga coherencia.

Una vez propuestas estas soluciones, es posible desde LabVIEW programar un modelo que cumpla las mismas para observar los resultados en una simulación *software*. En la Figura 4-30 se muestra la configuración escogida mediante componentes de bajo nivel.

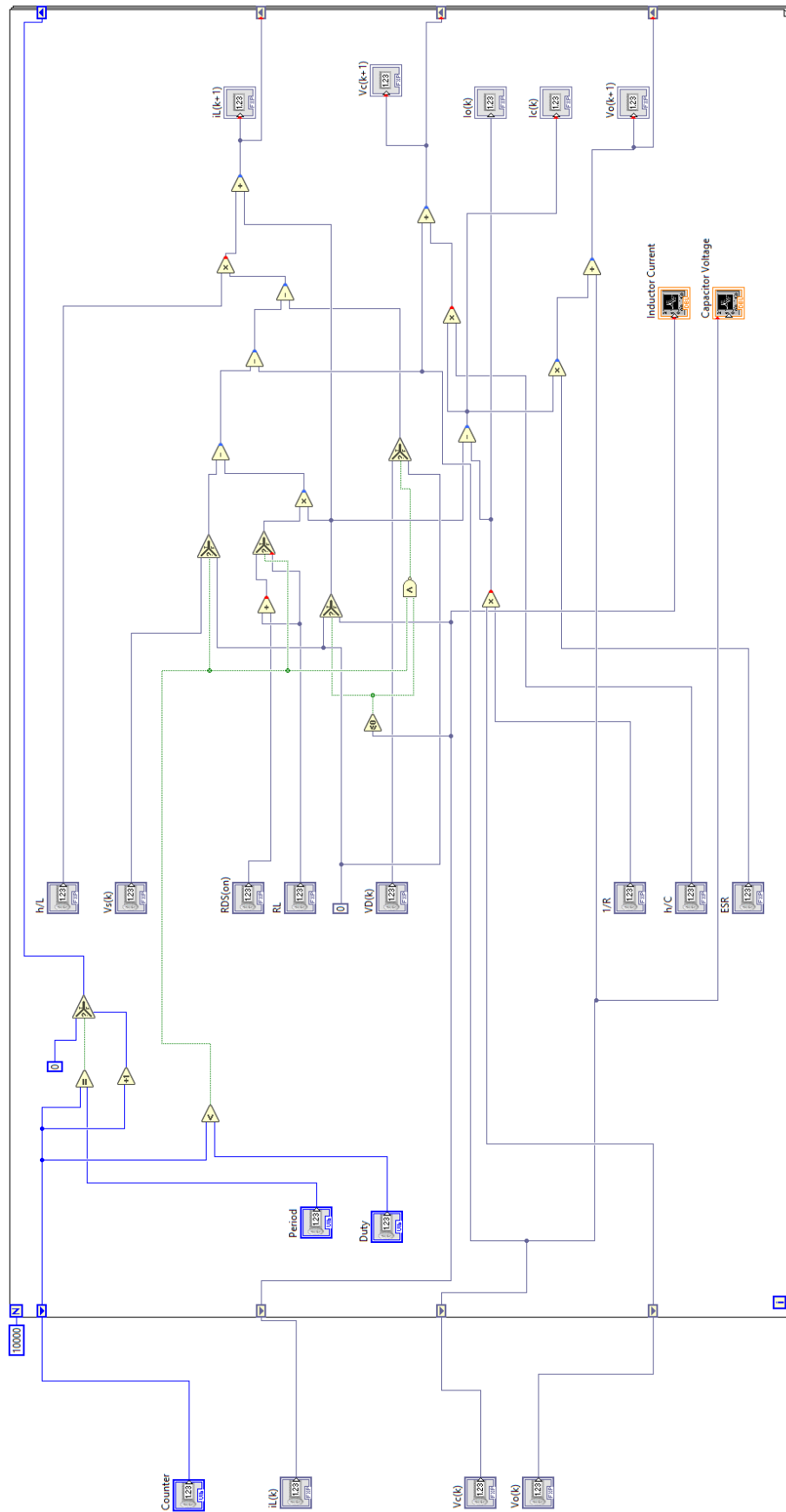


Figura 4-30: Diagrama de bloques simulación ecuaciones diferenciales *buck* en LabVIEW

Como se puede observar, toda la lógica se encuentra incluida dentro de un gran bucle “*for*” que se ejecuta hasta llegar al número de ciclos que se le especifique. Las señales de entrada i_L , V_c y V_o entran en el bucle, pasan por toda la lógica y se guarda su valor mediante “*shift registers*” o registros de desplazamiento, cuyos valores serán usados en la siguiente iteración. El tipo de elementos escogidos para el diseño son sumadores, restadores, multiplicadores, comparadores, elementos de selección y demás componentes de bajo nivel. El camino azul consiste en la generación de la señal de PWM (en color verde) mediante un sencillo contador que se compara con el periodo en ciclos escogido y el ciclo de trabajo o *Duty* para sacar el pulso de PWM correspondiente en cada momento. Todas las señales de entrada están implementadas con controles para poder variar su valor posteriormente en el panel frontal, mientras que las salidas se sacan a indicadores para observar sus valores. De cara a la simulación, se sacan además tanto el voltaje de salida como la corriente por la bobina a sendas gráficas para poder visualizar su comportamiento.

En la Figura 4-31 se muestra el panel frontal del modelo diseñado con un ejemplo de funcionamiento mediante una simulación sin pérdidas del sistema *buck* propuesto.

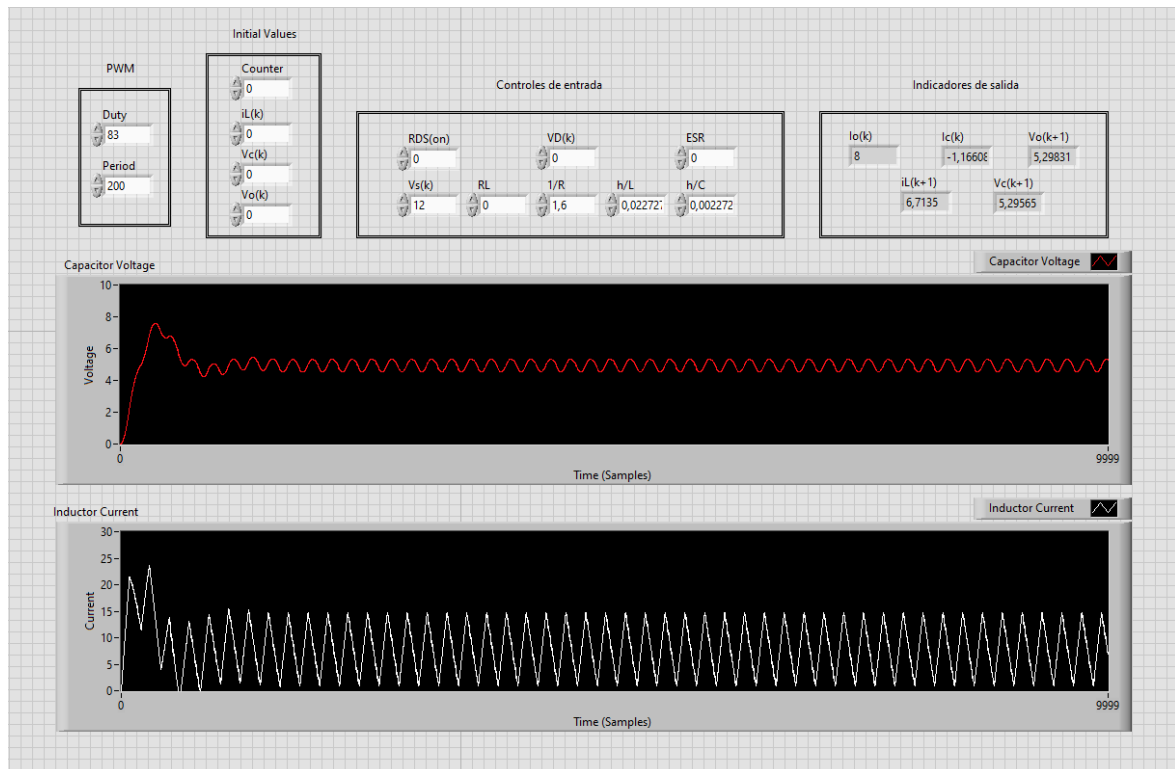
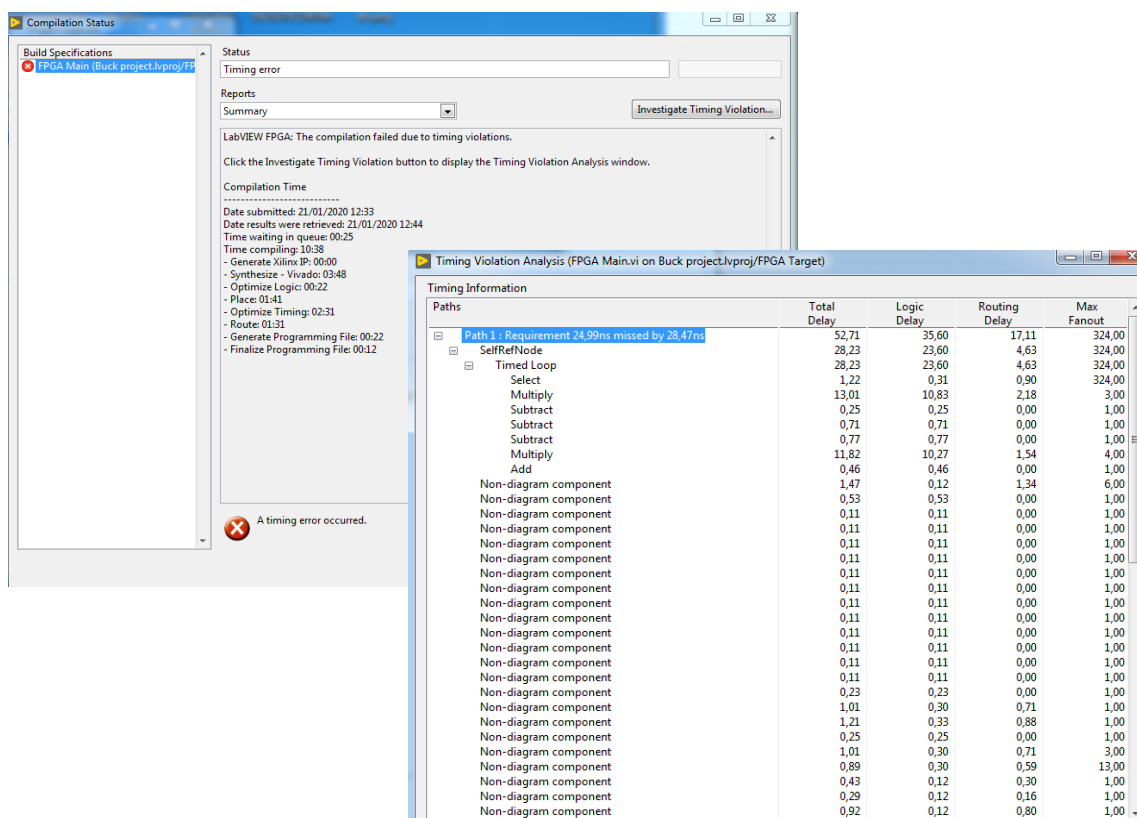


Figura 4-31: Panel frontal simulación ecuaciones diferenciales *buck* en LabVIEW

Sin embargo, esta simulación no se trata en ningún caso de una simulación HIL, ya que es simplemente una simulación *software* de la lógica de las ecuaciones diferenciales. Si se trata de desplegar directamente este modelo a la placa seleccionada van a suceder dos problemas.

El primero de ellos consiste en que el sistema myRIO-1900 con FPGA Zynq-7000 no dispone de osciloscopio integrado, luego no es posible emplear el módulo de “*Waveform*” para observar directamente los resultados. Para resolver este inconveniente, será necesario

El segundo viene ligado a temas de temporización o *timing*, y es que resulta que, para cumplir las especificaciones temporales deseadas, es decir, para que se realicen todos los cálculos deseados en cada ciclo, no basta con este tipo de elementos a bajo nivel ya que no son capaces de satisfacer estas condiciones. En la Figura 4-32 se muestra el error de *timing* obtenido al intentar desplegar el sistema sobre la FPGA directamente que, como se puede apreciar, está especialmente perjudicado por las operaciones de multiplicación



87

se permite configurar el modo de ejecución de estos bloques en función de si el elemento va a estar dentro o fuera de un bucle de ejecución en un único ciclo. Todos los bloques en este diseño estarán dentro del bucle principal.

Junto a dichos bloques DSP de alto rendimiento se emplearán además nodos de realimentación o “*Feedback Nodes*” que actúan a modo de bloque de retardo z^{-1} para retardar y sincronizar la lógica del circuito.

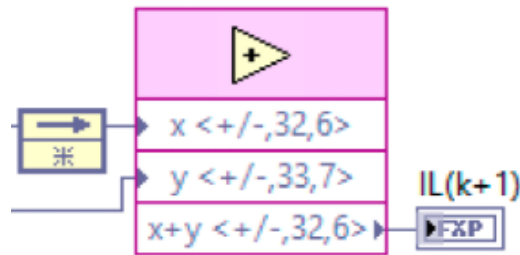


Figura 4-33: Ejemplo Feedback Node y Bloque Sumador High-Throughput DSP LabVIEW FPGA [6]

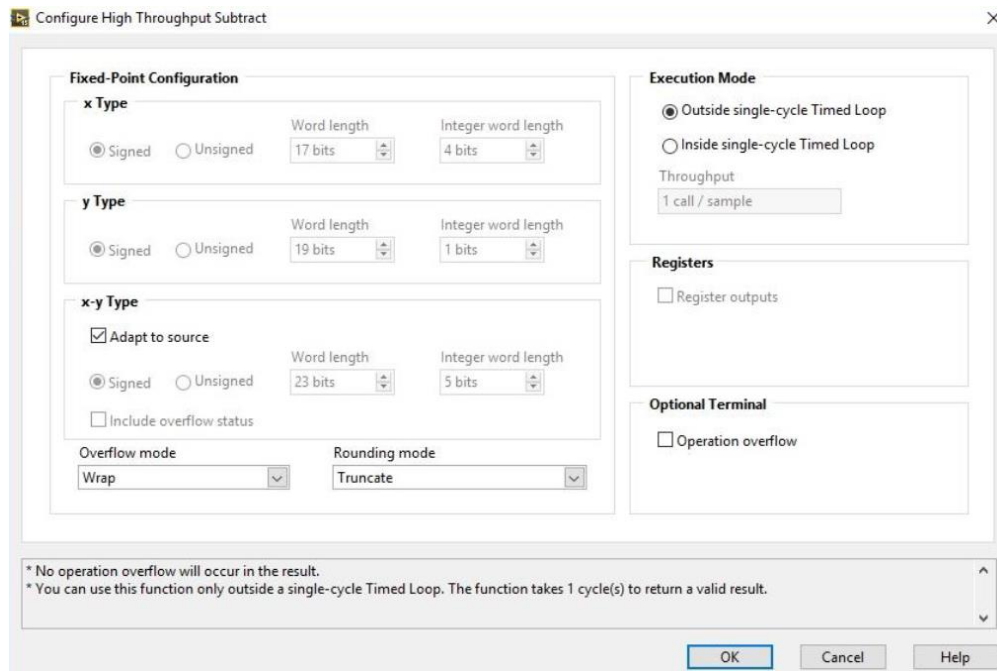


Figura 4-34: Ejemplo configuración High-Throughput DSP LabVIEW FPGA

A continuación, se va a proceder a enumerar los cuatro bloques principales diseñados en el diagrama de bloques de LabVIEW, que se ejecutan de forma paralela:

- Bloque de generación de PWM:

Sirve para generar la señal PWM a partir del periodo y *Duty* indicados en número de ciclos. En la Figura 4-35 se observa su aspecto.

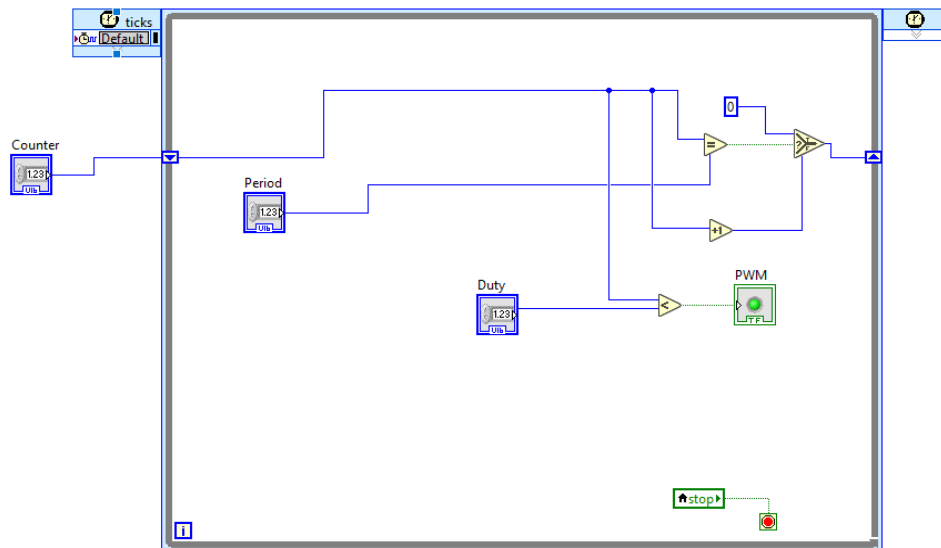


Figura 4-35: Bloque generación PWM LabVIEW FPGA

- Bloque de cálculo del Loop Rate:

Bloque interno al bucle principal que da a la salida en todo momento el “Loop Rate” o la tasa del bucle, que es el número de ciclos que tarda dicho bucle principal en ejecutarse. El valor mínimo de ciclos que tarda el modelo en producir cada solución es de 6 ciclos, lo que con un reloj de 40 MHz se traduce en 150 ns por solución. En la Figura 4-36 se muestra su diseño.

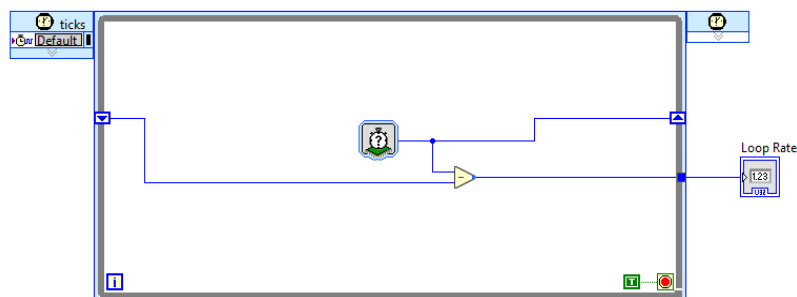


Figura 4-36: Bloque cálculo Loop Rate LabVIEW FPGA

- Módulo de sincronismo del paso de simulación:

Implementación de un sistema de sincronismo a través de la espera de un número acordado de ciclos mediante “wait”, para amoldarse a distintos pasos de simulación, en particular a los 500 ns de Typhoon.

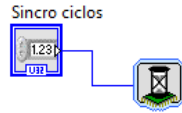


Figura 4-37: Módulo sincronismo LabVIEW FPGA

- Bloque de escritura en pines de salida:

Bloque mediante el cual se digitaliza el voltaje de salida o la corriente por la bobina calculados y se escribe en un pin analógico. Para este caso se escogieron las salidas A0 del conector MSP C ya que se va a trabajar con valores de tensión de salida de unos 5 V, que es precisamente la máxima salida que permiten los demás conectores, mientras que el C permite tensiones de hasta 10 V, lo que permite dejar un mayor rango de seguridad. Por otro lado, como se muestra en la documentación de la placa myRIO-1900 [13], al valor analógico se le escala por $1 / (4,883 \text{ mV})$, es decir por 205, para obtener la lectura adecuada en cada caso. Las fórmulas de conversión se muestran a continuación, mientras que el modelado del bloque se detalla en la Figura 4-38.

$$\text{Peso bit menos significativo} = 20 \text{ V} \div 2^{12} = 4,883 \text{ mV}$$

$$\text{Máxima lectura positiva} = +2047 \times 4,883 \text{ mV} = 9,995 \text{ V}$$

$$\text{Máxima lectura negativa} = -2048 \times 4,883 \text{ mV} = -10 \text{ V}$$

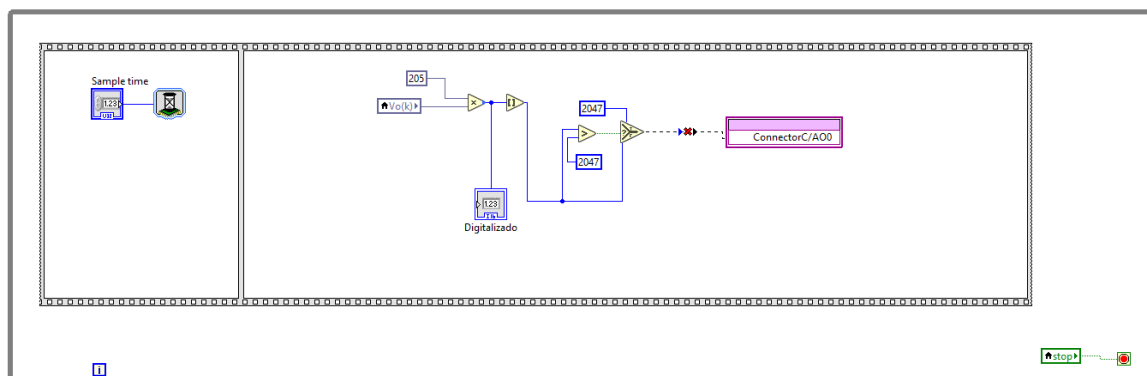


Figura 4-38: Bloque de escritura de salida LabVIEW FPGA

- Bucle principal:

Bucle en el que se lleva a cabo toda la lógica de las ecuaciones en diferencias amoldándose al empleo de bloques DSP de alto rendimiento, configurados individualmente para adaptarse al formato de coma fija óptimo para cada operación, y nodos de realimentación. En la Figura 4-39 se muestra todo el bucle “*while*” principal dentro del cual, como se puede observar, está tanto el módulo de sincronización de tipo “*wait*” como el bloque de cálculo del “*Loop Rate*”. Tanto la generación del PWM como la escritura de salida de hacen de manera paralela, pero externas a dicho bucle.

Desde el panel frontal se deberá dar el valor correspondiente a las distintas señales para lograr el funcionamiento del *buck* descrito en el apartado 4.1. Para ello los valores a dar a las señales, para el caso sin pérdidas, se muestran en la Tabla 4-6.

Parámetro	Valor	Descripción
V_s	12 V	Voltaje de entrada
h (dt)	500 ns	<i>Time Step</i>
L	22 μ H	Valor bobina
C	220 μ F	Valor condensador
h/L (dt/L)	0,0227272	Constante modelo
h/C (dt/C)	0,00227272	Constante modelo
R	0,625 Ω	Resistencia carga
1/R	1,6 S	Inverso carga
Periodo	200 ciclos	Periodo PWM
Duty	83 ciclos	Ciclo de trabajo
ESR	0	Resistencia serie equivalente del condensador
R_L	0	Resistencia serie de la bobina
R_{DS,on}	0	Resistencia del interruptor en estado ON
V_d	0	Caída voltaje diodo
Valores iniciales	0	Voltaje y corriente iniciales

Tabla 4-6: Parámetros panel frontal *buck* LabVIEW FPGA (sin pérdidas)

Bucle principal

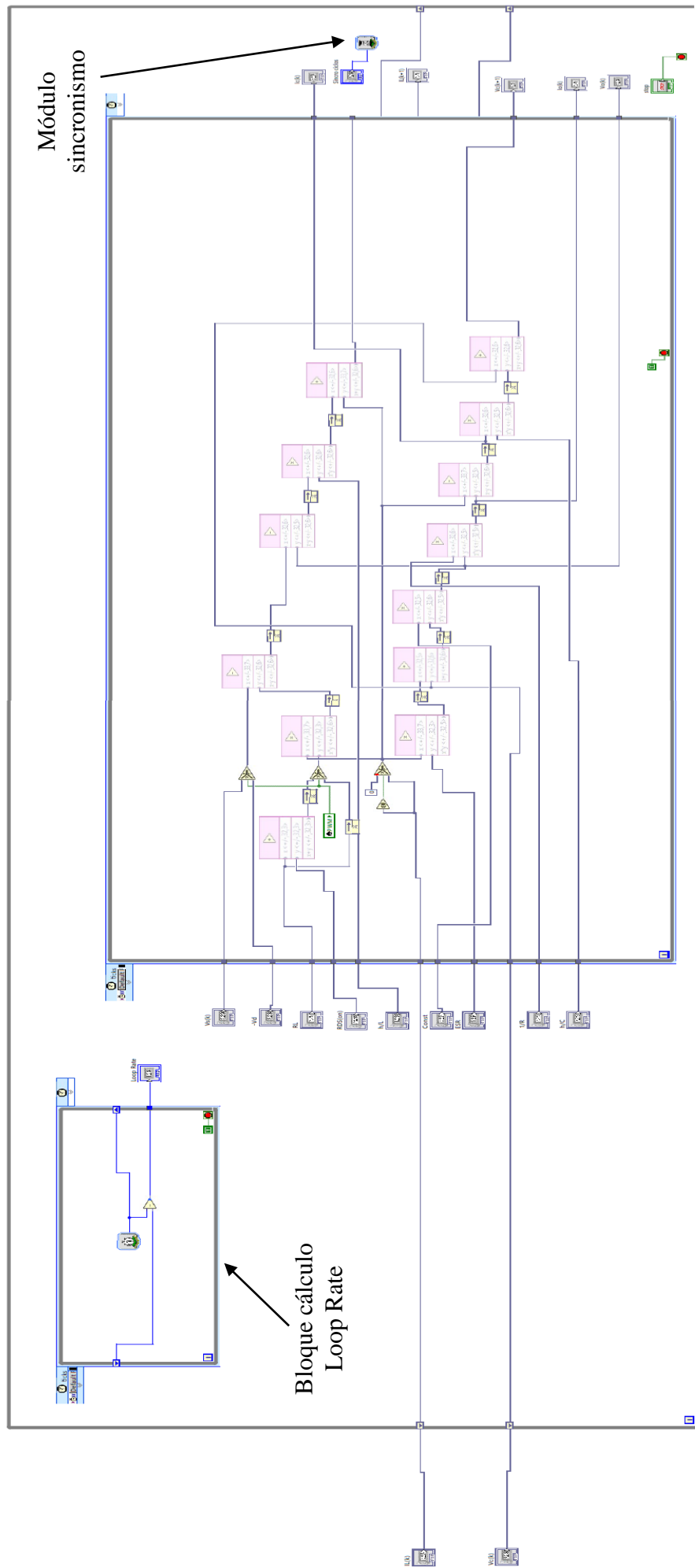


Figura 4-39: Bucle principal LabVIEW FPGA

El periodo de PWM en ciclos se calcula teniendo en cuenta que se dispone en la placa de un reloj de 40 MHz y se quiere un PWM a una frecuencia de 200 kHz, luego:

$$\frac{40 \text{ MHz}}{200 \text{ kHz}} = 200 \text{ ciclos}$$

A partir de dicho valor, ciclo de trabajo de $5/12 = 0,41667$ se calcularía multiplicando por dicho periodo y redondeando:

$$200 \text{ ciclos} \times 0,41667 = 83,33 \text{ (redondeo a 83 ciclos)}$$

En la Figura 4-40 se ilustra como quedaría el panel frontal durante una ejecución normal del modelo.

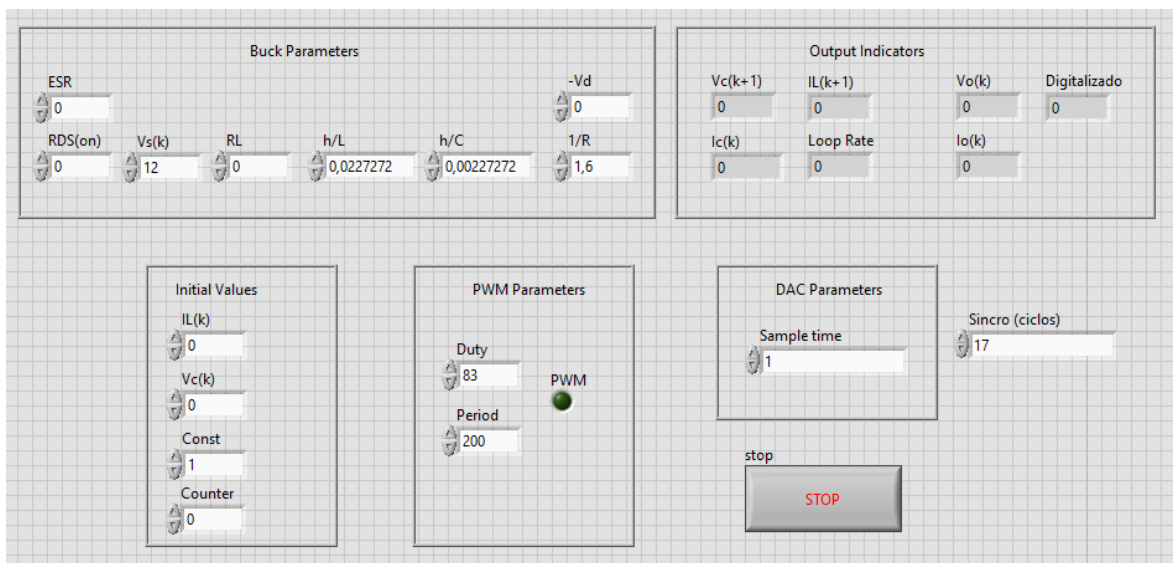


Figura 4-40: Panel frontal *buck* LabVIEW FPGA (sin pérdidas)

Una vez diseñado, se procede a compilar el modelo utilizando una de las opciones de compilación que permite el *software* [23]. En el caso de este trabajo se ha empleado la compilación desde el servidor de LabVIEW.

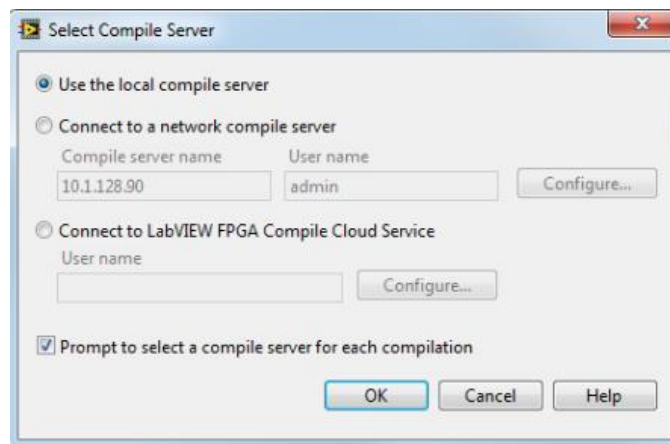


Figura 4-41: Métodos de compilación de LabVIEW [23]

De este modo se sintetiza el circuito y se generan los archivos *bit files* necesarios para bajarlo a la placa seleccionada. Ya sin errores ni violaciones de *timing* como los mostrados en la Figura 4-32, aparece la utilización estimada durante la fase de síntesis y, posteriormente, la utilización final en la fase de *placement* como aparece en la Figura 4-42. Estos reportes aparecen en términos de componentes como registros, LUTs, block RAMs y DSPs y, como era de esperar, son diferentes en cada fase. A la vista de estos resultados, se puede inferir que solo en un *buck* diseñado de esta forma el total de recursos *slice* empleados es del 78 % y el de DSP del 40 %, por lo que sería imposible desplegar varios *bucks* simultáneamente como sí permite la máquina HIL402 de Typhoon HIL.

Status

Complete

Reports

Estimated device utilization (synthesis)

Device Utilization	Used	Total	Percent
Slice Registers	11746	35200	33,4
Slice LUTs	11667	17600	66,3
Block RAMs	5	60	8,3
DSP48s	32	80	40,0

Status

Complete

Reports

Final device utilization (placement)

Device Utilization	Used	Total	Percent
Total Slices	3432	4400	78,0
Slice Registers	10679	35200	30,3
Slice LUTs	10000	17600	56,8
Block RAMs	5	60	8,3
DSP48s	32	80	40,0

Figura 4-42: Utilización en fase de síntesis y *placement* circuito *buck* LabVIEW FPGA

Una vez hecho el despliegue, se encuentra el modelo ejecutándose sobre la placa a tiempo real, pudiendo variar en cualquier momento los valores de los controles de entrada del panel frontal para variar los resultados de salida. En la Figura 4-43 se muestra un montaje típico del despliegue de este circuito realizado en el laboratorio del HCTLab de la Escuela Politécnica de Madrid, en el que se muestra un escalón de salida en el osciloscopio.

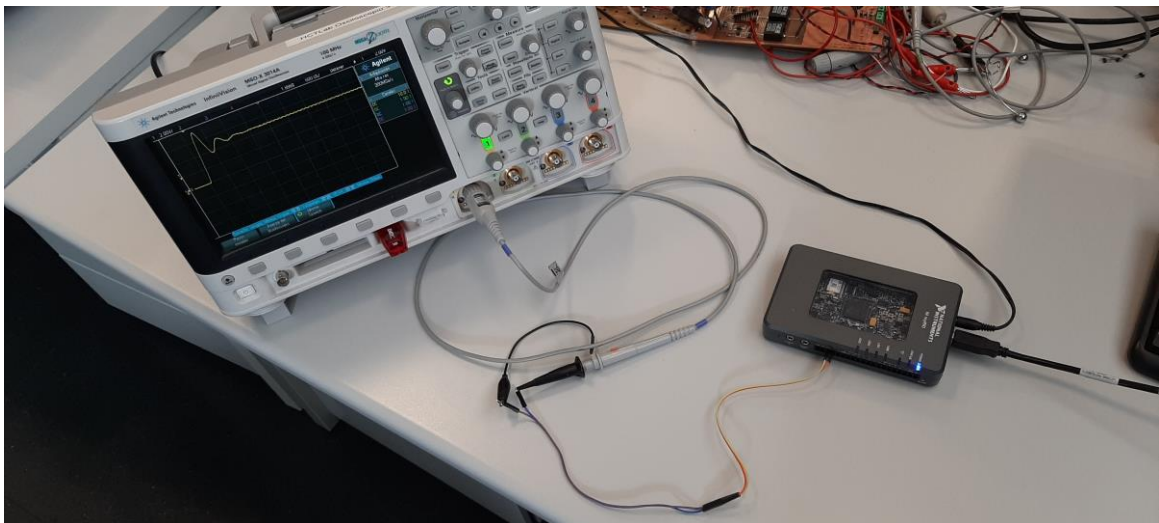


Figura 4-43: Montaje circuito *buck* LabVIEW FPGA

4.3 Diseño, simulación y montaje convertidor conmutado buck (con pérdidas)

Como se ha comprobado en apartados anteriores, en especial en el 4.2.2, mientras que en Simulink y en el modelo diseñado en LabVIEW es posible introducir todas las pérdidas que se buscaban en un principio, Typhoon HIL obliga a emplear su propio bloque de convertidor *buck*, carente de la posibilidad de introducir pérdidas en su diodo (V_D o R_D). Tampoco es viable la opción de recrear un convertidor *buck* mediante componentes sueltos (diodo y transistor), ya que Typhoon es muy específico en ese sentido y ofrece únicamente bloques complejos de más alto nivel ya integrados. Además, aunque este *software* permite la creación de máscaras para añadir otras funcionalidades en ciertos componentes, como bobinas o condensadores, no es posible acceder a la edición de máscara del bloque *buck* ya que se encuentra bloqueado.

Por esto, nuevamente Typhoon será limitante en este aspecto y, por ello, las pérdidas a modelar en cada circuito serán finalmente las siguientes:

- Resistencia *Drain-Source*: $R_{DS_{ON}} = 0,02 \, \Omega$
- Resistencia bobina: $R_L = 0,02 \, \Omega$
- Resistencia equivalente serie: $ESR = 0,1 \, \Omega$

4.3.1 Modelo teórico

Para realizar el modelo teórico del convertidor *buck* en Simulink con pérdidas, se ha usado prácticamente la misma configuración que la ya descrita en el apartado 4.2.1, con la diferencia de que se han introducido los elementos correspondientes para simular las pérdidas deseadas. En la Figura 4-44 se muestra el circuito simulado con pérdidas.

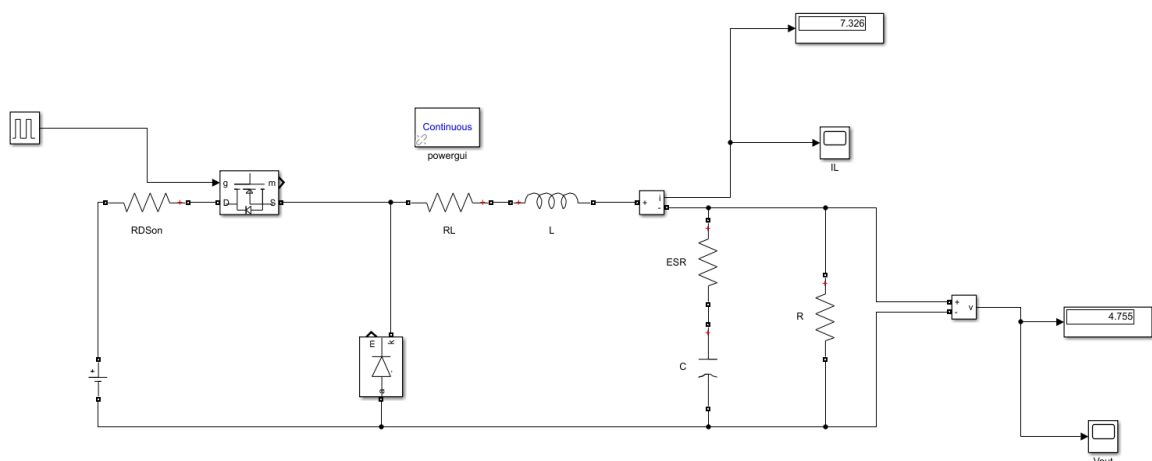


Figura 4-44: Simulación *buck* teórico Simulink (con pérdidas)

Para el caso con pérdidas, las *waveforms* obtenidas para ambas medidas a comparar (I_L y V_O) aparecen representadas en las Figuras 4-45 y 4-46.

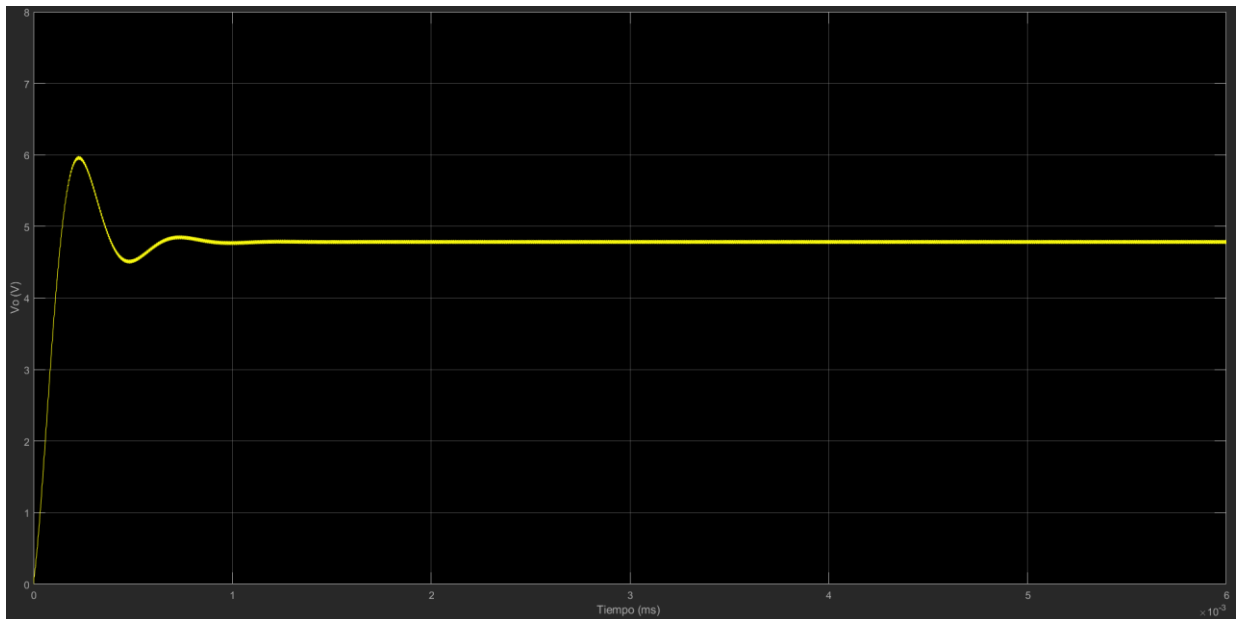


Figura 4-45: Medida V_O teórica Simulink (con pérdidas)

En términos de voltaje, para el caso con pérdidas, se observa una menor oscilación, que apenas alcanza los 6 V en su mayor pico de sobreoscilación. Además, el voltaje en régimen permanente es de unos 4,78 V, no llegando a alcanzar los 5 V del caso sin pérdidas.

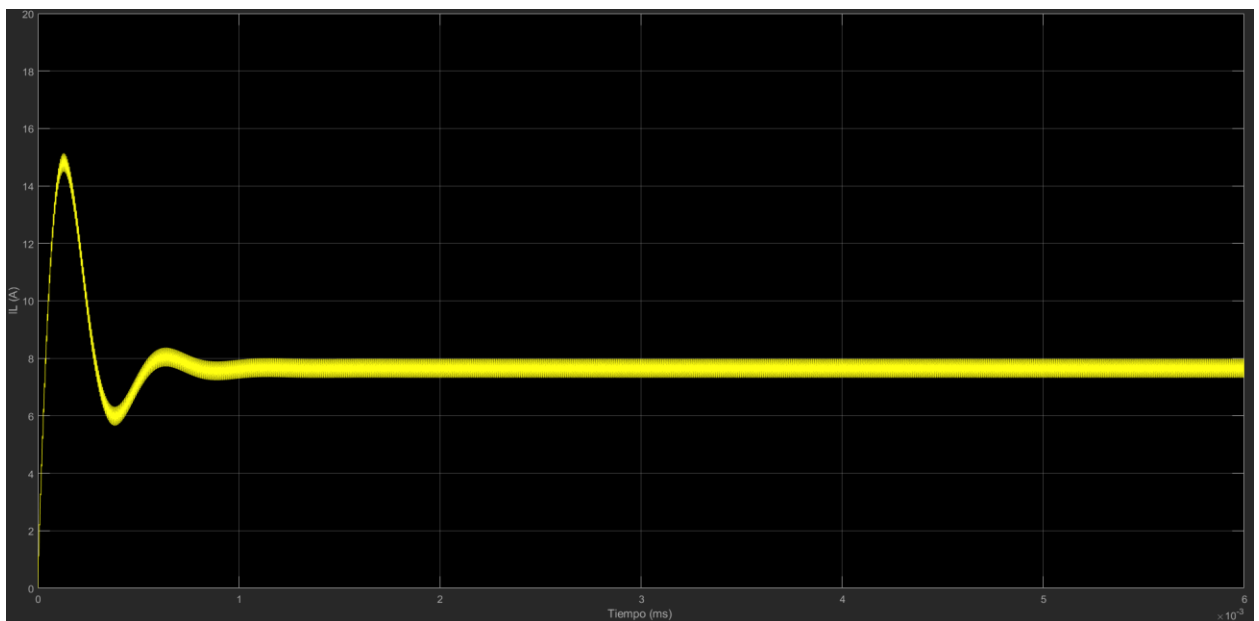


Figura 4-46: Medida I_L teórica Simulink (con pérdidas)

Para la corriente sucede algo parecido, con una sobreoscilación de 15 V de máximo y una corriente media en régimen permanente de 7,65 V, no llegando a los 8 V, y presenta el mismo rizado de 0,66 A que en el modelo sin pérdidas.

4.3.2 Typhoon HIL

Para el diseño con pérdidas en la plataforma Typhoon HIL se parte del circuito implementado en el apartado 4.2.2 y se realizan sobre éste las modificaciones oportunas. Un detalle a tener en cuenta en este apartado consiste en las limitaciones en cuanto a las pérdidas que Typhoon HIL permite incluir en este tipo de convertidores. En la Figura 4-11 se muestran los posibles parámetros a manipular del bloque *buck* diseñado por la empresa, entre los que no aparecen ni el voltaje del diodo (V_D) ni su resistencia (R_D). Como se anticipaba en secciones anteriores del trabajo, Typhoon HIL es muy específico en cuanto al tipo de componentes y bloques que se permiten añadir a un circuito. Normalmente hay bloques de alto nivel ya integrados y no se ofrece la posibilidad de añadir, por ejemplo, un diodo o un transistor para poder hacer el diseño del *buck* a bajo nivel para poder añadir todas las pérdidas definidas en un principio en el apartado 4.1.

Otra posibilidad consiste en la definición manual de “máscaras” para ciertos componentes, que permite añadir características personalizadas a los mismos. El tema es que, en determinados bloques, como es el caso del *buck*, esta opción se encuentra bloqueada y no es posible alterar el comportamiento del bloque internamente.

Por ello, y ante estas limitaciones, las pérdidas finales incorporadas son las expuestas en el apartado anterior y las modificaciones sobre el circuito inicial se pueden observar sobre la Figura 4-47.

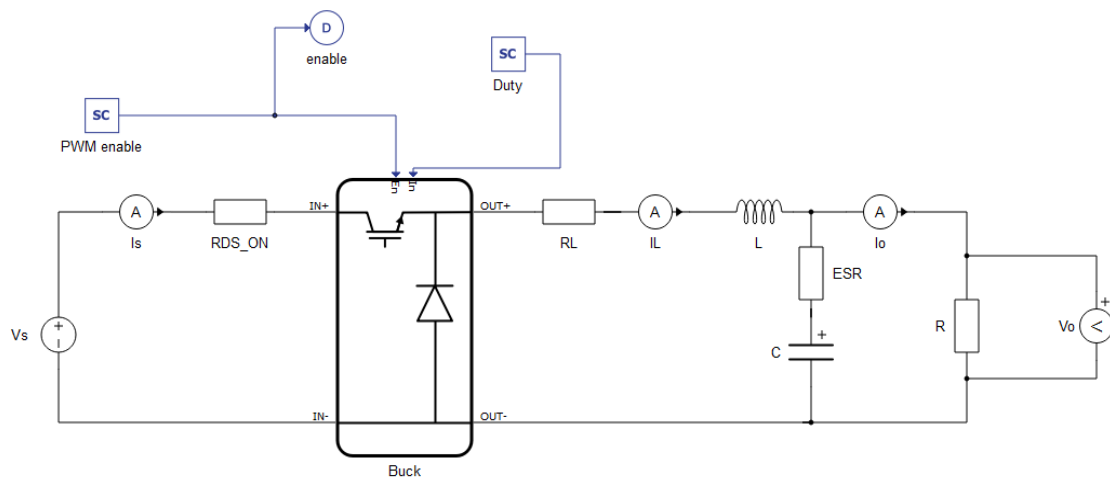


Figura 4-47: Circuito *buck* Typhoon HIL (con pérdidas)

Usando exactamente los mismos parámetros que en la simulación inicial (mismo tipo de *solver*, paso de simulación, frecuencia de conmutación, *Duty*, etc.) se procede a compilar el sistema.

En cuanto a la utilización, el hecho de añadir las resistencias de pérdidas no tiene efecto significativo en este aspecto y se consiguen los mismos resultados que los mostrados sobre las tablas 4-1, 4-2 y 4-3.

Cargando el sistema compilado en HIL SCADA y, empleando el mismo panel que en el caso sin pérdidas, se procede a capturar el voltaje de salida y la corriente por la bobina para esta nueva configuración. En la Figura 4-48 se muestra la captura de las señales V_s , V_o , I_s e I_L .

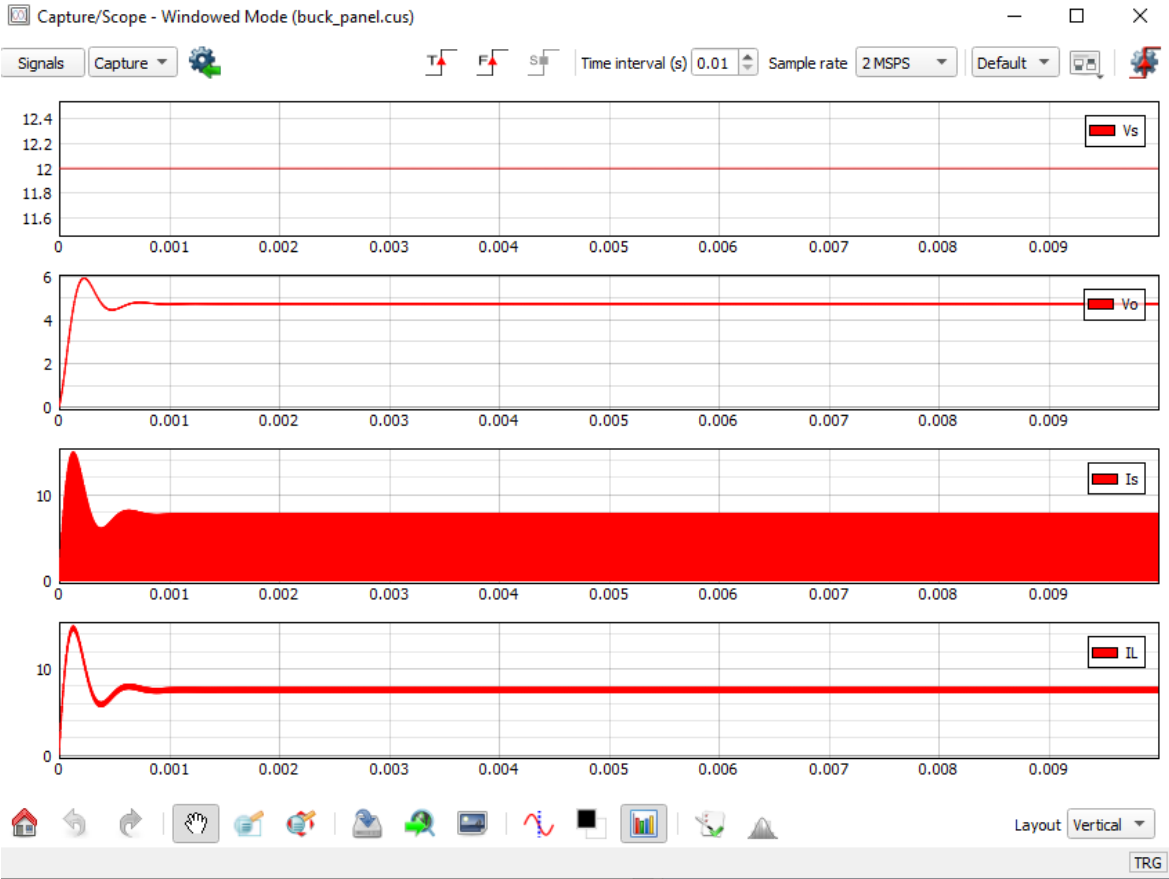


Figura 4-48: Simulación Capture buck HIL SCADA (con pérdidas)

En la Figura 4-49 se muestra la medida para todas estas señales de un cursor posicionado sobre un punto determinado del régimen permanente.

Signal Name	Cursor 1
▼ Viewport 1	
⌚ Time	0.005s
■ Vs	12.0
▼ Viewport 2	
⌚ Time	0.005s
■ Vo	4.715257
▼ Viewport 3	
⌚ Time	0.005s
■ Is	7.392268
▼ Viewport 4	
⌚ Time	0.005s
■ IL	7.392268

Figura 4-49: Cursor simulación Capture buck HIL SCADA (con pérdidas)

Como se puede observar, las señales se asemejan bastante a las del caso ideal con pérdidas tanto en forma como en valores y tiempos.

4.3.3 LabVIEW FPGA

El diseño es exactamente idéntico al realizado en el apartado 4.2.4 para el caso sin pérdidas ya que desde el primer momento se ha trabajado con las variables de pérdidas correspondientes de cara a poder añadirlas más tarde y, además, se encuentra todo parametrizado. En este caso, pese a poder añadir pérdidas tales como el voltaje por el diodo, se añaden solamente las mismas que permite Typhoon HIL de las que se propusieron en un primer momento, es decir: ESR, R_L y $R_{DS,on}$.

Parámetro	Valor	Descripción
V_s	12 V	Voltaje de entrada
h (dt)	500 ns	<i>Time Step</i>
L	22 μ H	Valor bobina
C	220 μ F	Valor condensador
h/L (dt/L)	0,0227272	Constante modelo
h/C (dt/C)	0,00227272	Constante modelo
R	0,625 Ω	Resistencia carga
$1/R$	1,6 S	Inverso carga
Periodo	200 ciclos	Periodo PWM
Duty	83 ciclos	Ciclo de trabajo
ESR	0,1 Ω	Resistencia serie equivalente del condensador
R_L	0,02 Ω	Resistencia serie de la bobina
$R_{DS,on}$	0,02 Ω	Resistencia del interruptor en estado ON
V_d	0	Caída voltaje diodo
Valores iniciales	0	Voltaje y corriente iniciales

Tabla 4-7: Parámetros panel frontal *buck* LabVIEW FPGA (con pérdidas)

A la hora de simular en el panel frontal deben quedar reflejados dichos parámetros como se muestra en la Figura 4-50.

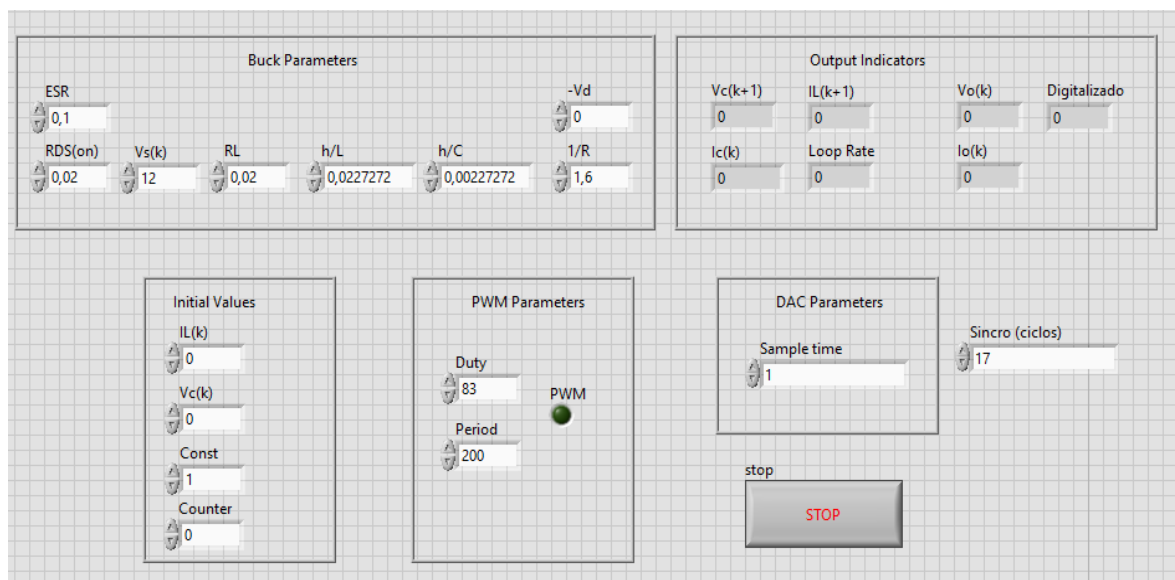


Figura 4-50: Panel frontal *buck* LabVIEW FPGA (con pérdidas)

4.4 Comparativa de resultados simulación HIL convertidor buck

En este apartado se procederá a mostrar la comparativa de los resultados obtenidos en el apartado anterior, de las simulaciones HIL del circuito *buck* propuesto en las tres plataformas: Simulink (modelo teórico), Typhoon HIL y LabVIEW FPGA. Según lo ya comentado, hay dos escenarios de pruebas: con pérdidas y sin pérdidas, y dentro de cada uno de ellos se observará tanto el voltaje de salida en la carga (V_O) como la corriente por la bobina (I_L), así como el rizado en ambos casos. Para todos estos casos, se adquirieron y exportaron los resultados a una serie de hojas Excel para su comparativa.

Aunque las simulaciones se hicieron con un paso de simulación o “dt” de 500 ns para que fuese equivalente en todas las plataformas, en el apartado 4.4.3 se mostrarán también los resultados de la simulación para LabVIEW con el mínimo “dt” posible en el modelo implementado, que es de 150 ns.

4.4.1 Buck asíncrono sin pérdidas

- Voltaje de salida (V_O)

En cuanto al voltaje de salida para el caso sin pérdidas, en la Figura 4-51 se muestran en la misma gráfica los resultados de Typhoon y LabVIEW. En la Figura 4-52 se muestra la misma gráfica con el transitorio ampliado y en la Figura 4-53 se añade a la comparativa el modelo ideal en Simulink de cara a tener la comparativa completa de los tres casos. Por último, en la Figura 4-54 se comparan los rizados obtenidos en las tres plataformas en una sección cualquiera del régimen permanente.

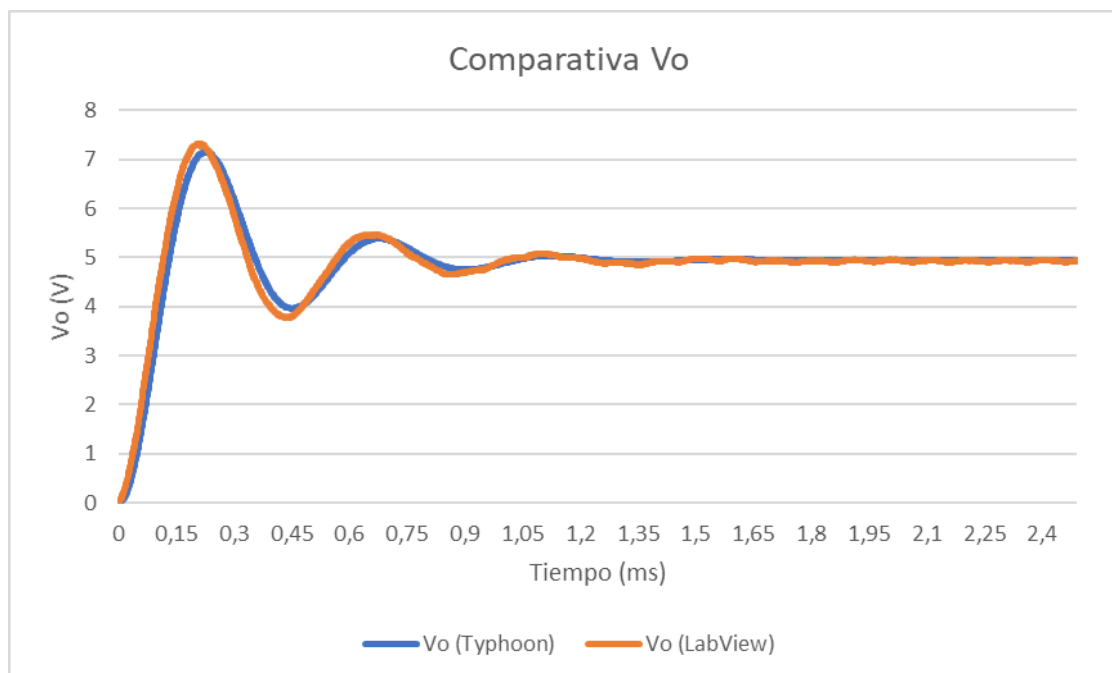


Figura 4-51: Comparativa V_O Typhoon vs LabVIEW (sin pérdidas)

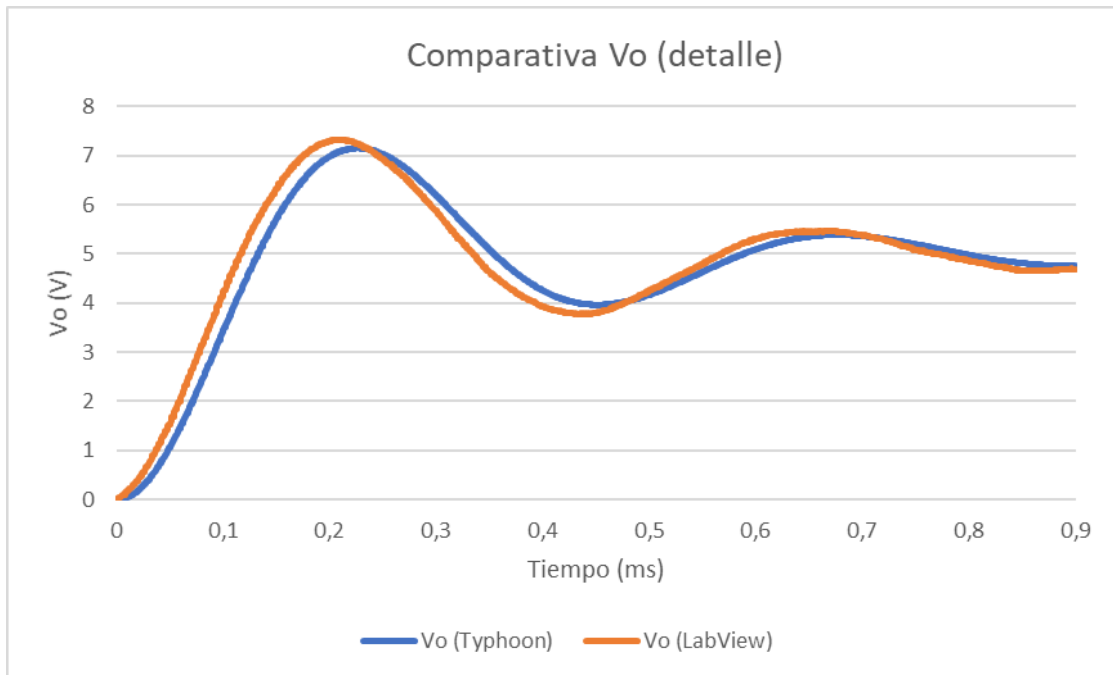


Figura 4-52: Detalle comparativa V_o Typhoon vs LabVIEW (sin pérdidas)

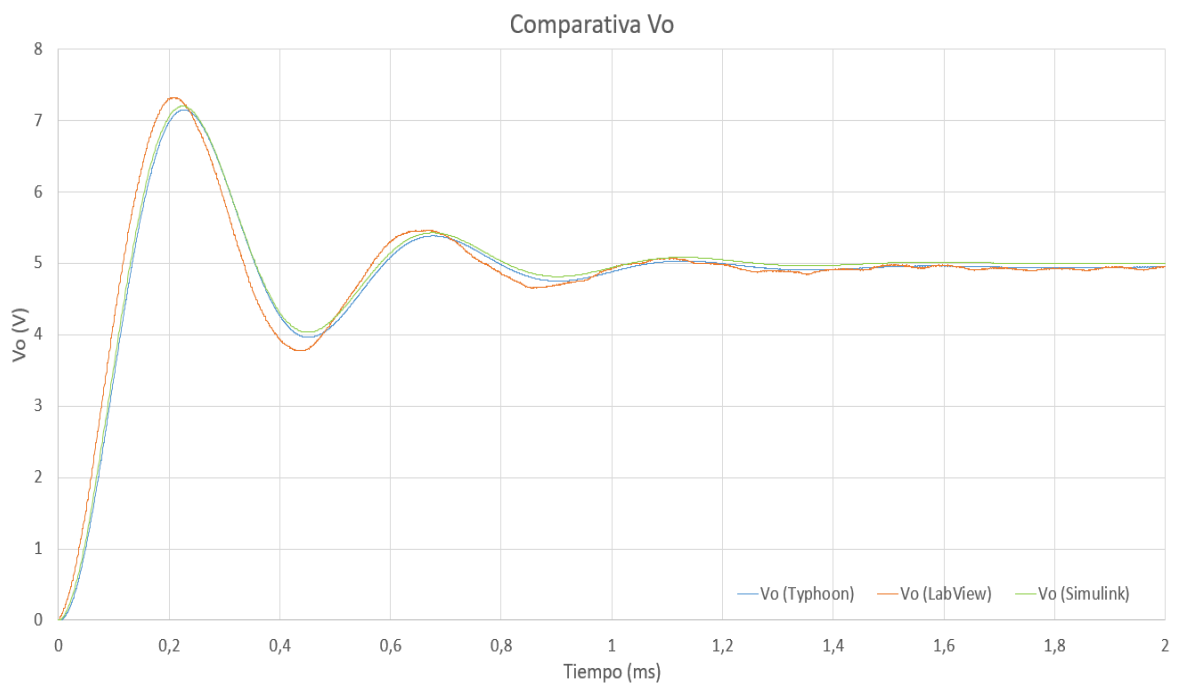


Figura 4-53: Comparativa V_o Typhoon vs LabVIEW vs Simulink (sin pérdidas)

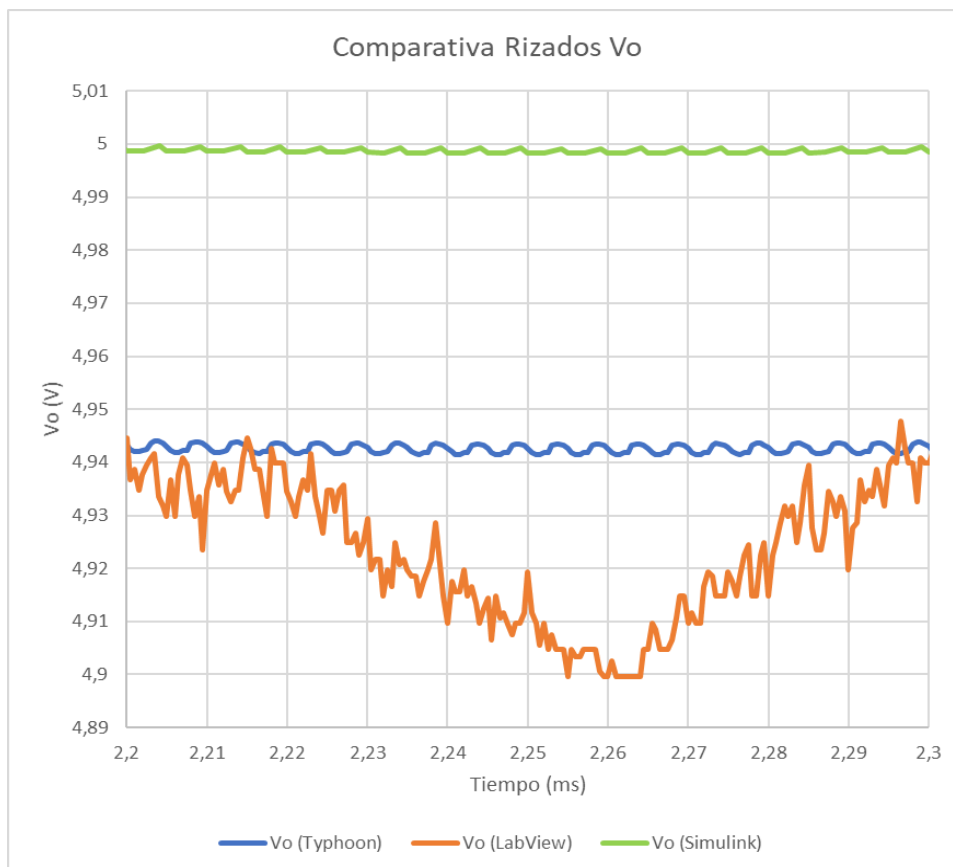


Figura 4-54: Comparativa rizados V_o Typhoon vs LabVIEW vs Simulink (sin pérdidas)

En líneas generales las dos plataformas HIL a comparar arrojan datos bastante similares entre ellas y muy parecidas a las del modelo ideal. En ambos casos se cumplen tiempos de establecimientos al 5% muy cercanos a los del modelo de Simulink de 0,74 ms, si bien es cierto que el circuito de LabVIEW FPGA va ligeramente retrasado temporalmente respecto a los otros dos. Uno de los temas más destacables es el tema de los rizados. Tal y como se aprecia en la Figura 4-54, el modelo teórico de Simulink presenta un levísimo rizado y llega sin problemas a los 5 V al contar con todas las idealidades de interruptores y pérdidas posibles. Typhoon HIL presenta también un rizado leve, pero en este caso no alcanza totalmente los 5 V, quedándose a tan solo unos 55 mV. Es para LabVIEW FPGA donde se aprecia una gran diferencia de comportamiento, donde además de un voltaje de salida más bajo se detecta un rizado periódico de unos 40 mV de amplitud mucho más abrupto que en el caso de Typhoon. Esto se puede explicar si se tiene en cuenta que este modelo es el único que se ha bajado físicamente a *hardware* (además a uno no especializado completamente en simulaciones HIL) y, posteriormente, se ha medido con un osciloscopio como se ha explicado en el apartado 4.2.4. Por ello, es el único modelo expuesto a ruido y, por si fuera poco, también se debe tener en cuenta el propio DAC del osciloscopio que altera y degrada la señal. Además, hay que recordar que la configuración diseñada en LabVIEW puede llegar hasta los 150 ns de paso de simulación y se ha forzado a 500 ns, retrasando los datos, de cara a adaptarse al limitante, que es Typhoon, lo que también puede influir en este aspecto. Pese a la ya comentada oscilación a causa de ruido y otros factores, el comportamiento de esta herramienta se puede decir que es bastante bueno, aunque no tan preciso como Typhoon HIL y también es menos intuitivo de implementar.

- **Corriente por la bobina (I_L)**

De forma similar para el caso de la corriente por la bobina en el escenario sin pérdidas, se muestran en las Figuras 4-55, 4-56 y 4-57 la comparativa entre los resultados de las tres herramientas y, en la Figura 4-58 se muestran los rizados de corrientes en un segmento del régimen permanente.

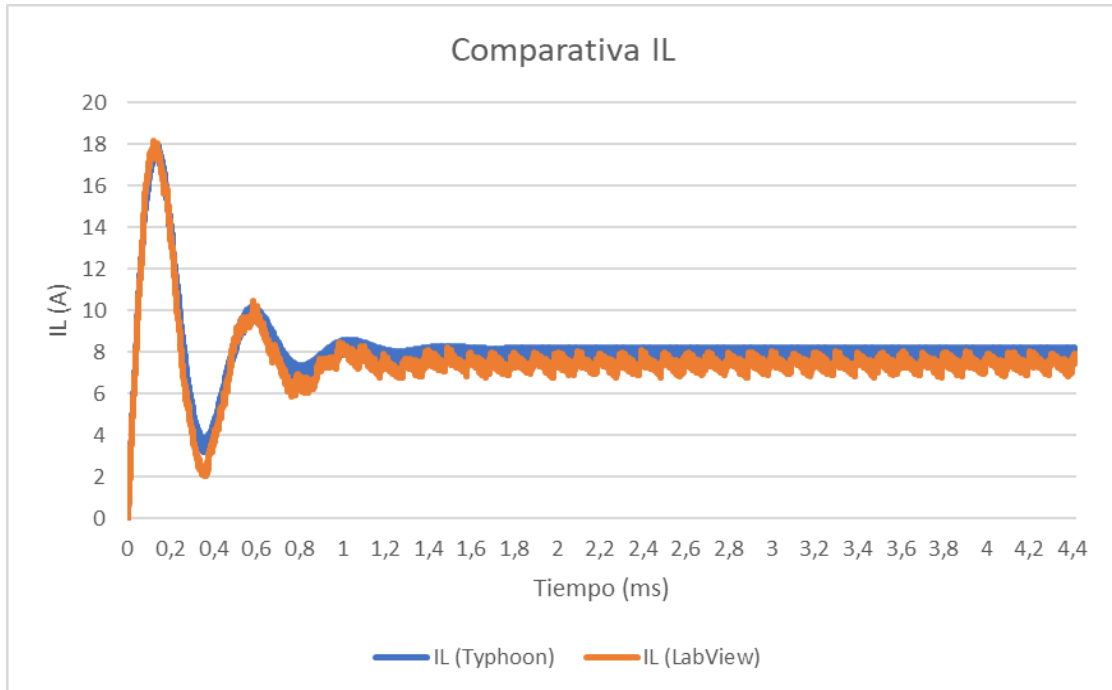


Figura 4-55: Comparativa I_L Typhoon vs LabVIEW (sin pérdidas)

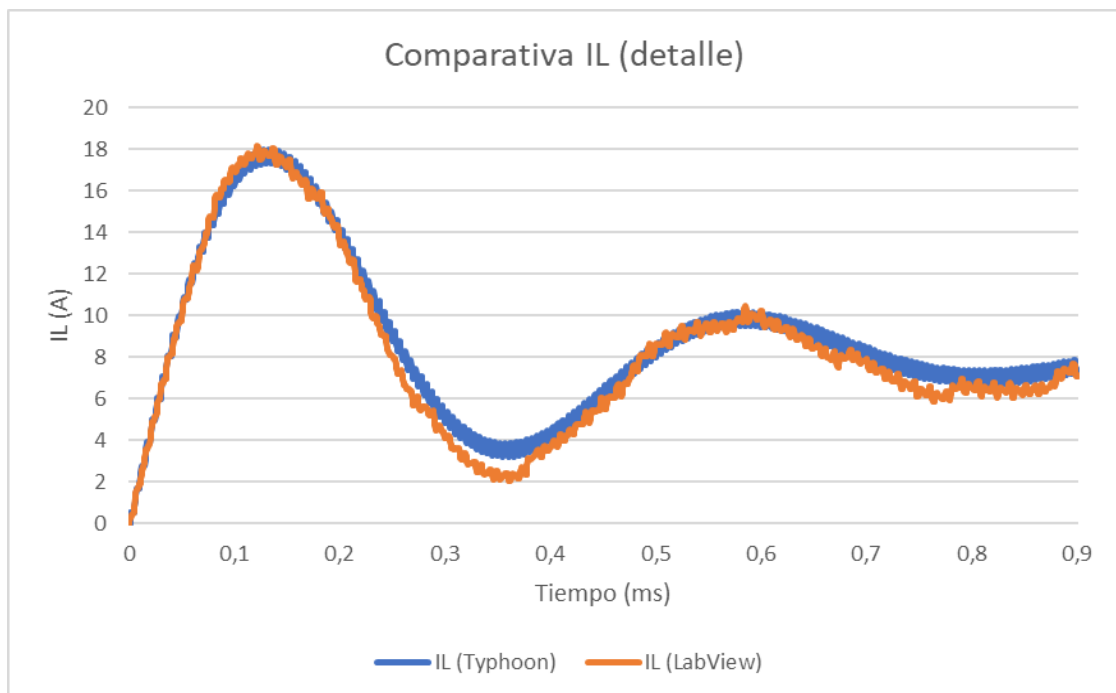


Figura 4-56: Detalle comparativa I_L Typhoon vs LabVIEW (sin pérdidas)

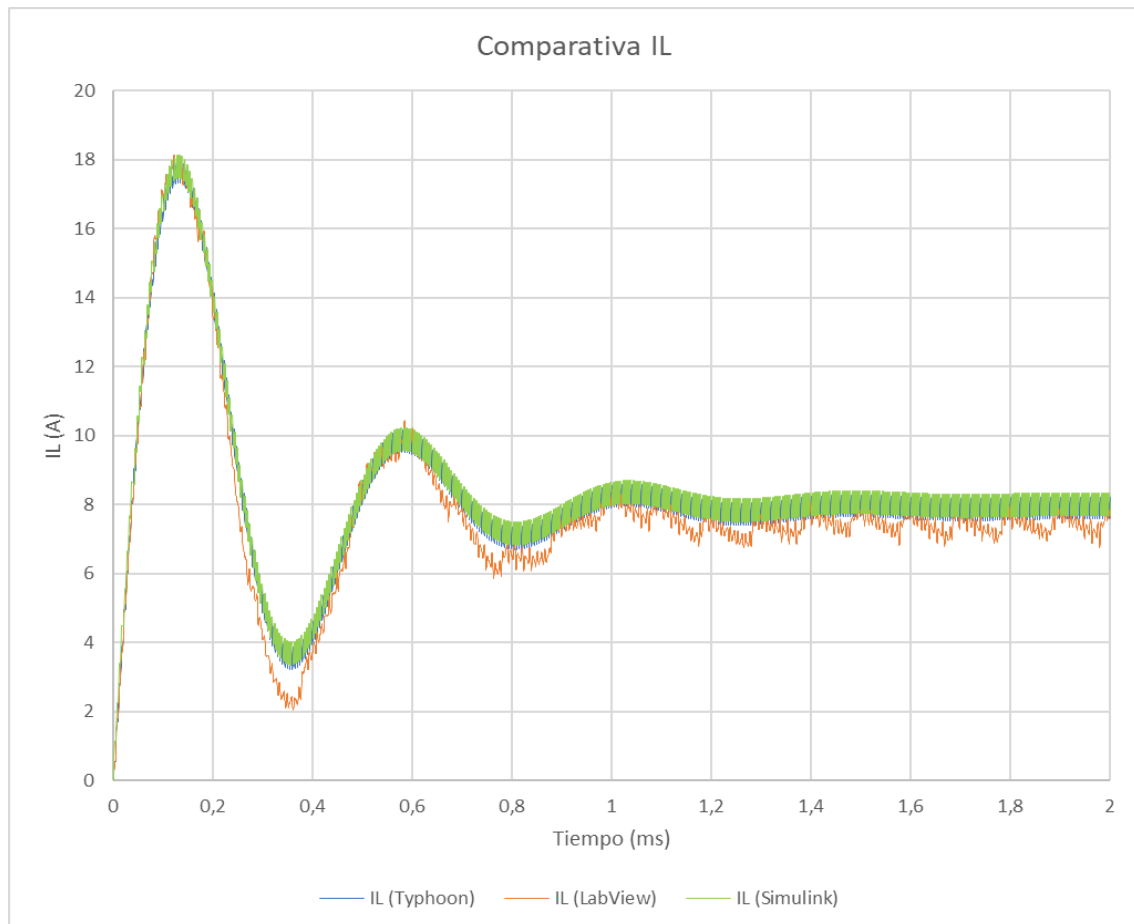


Figura 4-57: Comparativa I_L Typhoon vs LabVIEW vs Simulink (sin pérdidas)

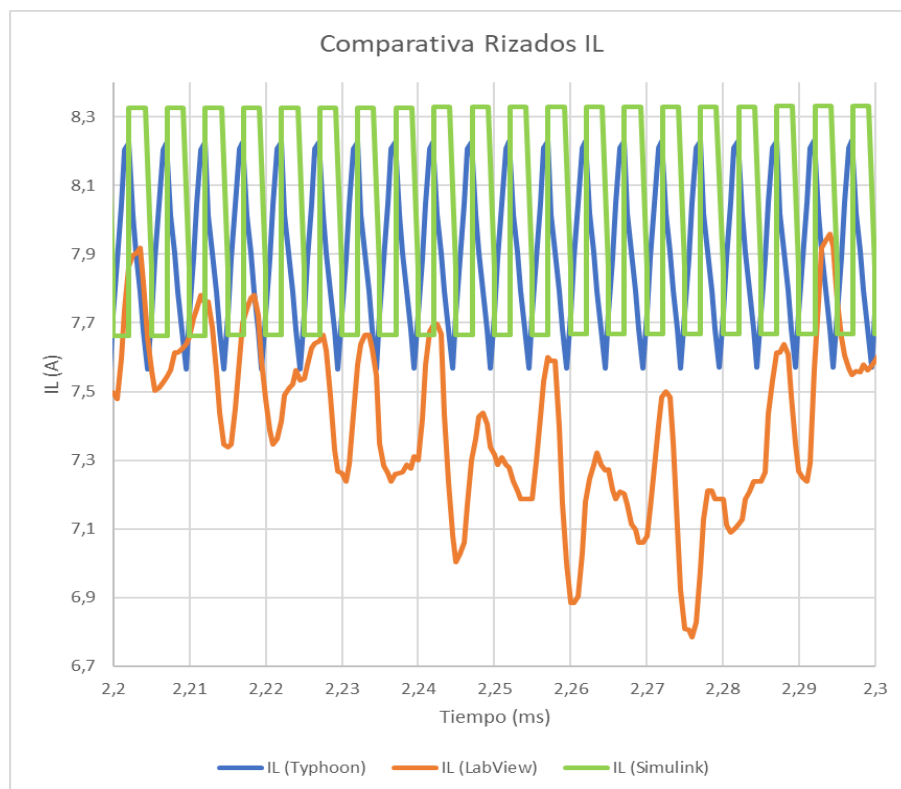


Figura 4-58: Comparativa rizados I_L Typhoon vs LabVIEW vs Simulink (sin pérdidas)

Nuevamente las dos simulaciones HIL comparadas arrojan similares resultados en cuanto a corrientes se refiere, ya que son similares entre sí y respecto al modelo teórico de Simulink. En este caso el circuito *buck* de Typhoon se comporta prácticamente igual que el caso ideal, cayendo apenas 0,1 A de amplitud respecto a éste. En cuanto a LabVIEW FPGA el efecto de ese pronunciado rizado que se veía para el voltaje es, aún si cabe, más abrupto para el caso de la corriente, pudiendo llegar a caer unos 0,6 A de corriente media. Nuevamente este efecto se explica por ser el único modelo sometido a ruido y pasado por el DAC de un osciloscopio, al haber sido montado y medido físicamente y temas de modelado.

4.4.2 Buck asíncrono con pérdidas

- Voltaje de salida (V_o)

La comparativa de resultados de voltaje de salida para el convertidor *buck* con pérdidas se puede observar en las Figuras 4-59, 4-60 y 4-61, y en la Figura 4-62 se muestra el rizado para las tres plataformas en una sección determinada del régimen permanente.

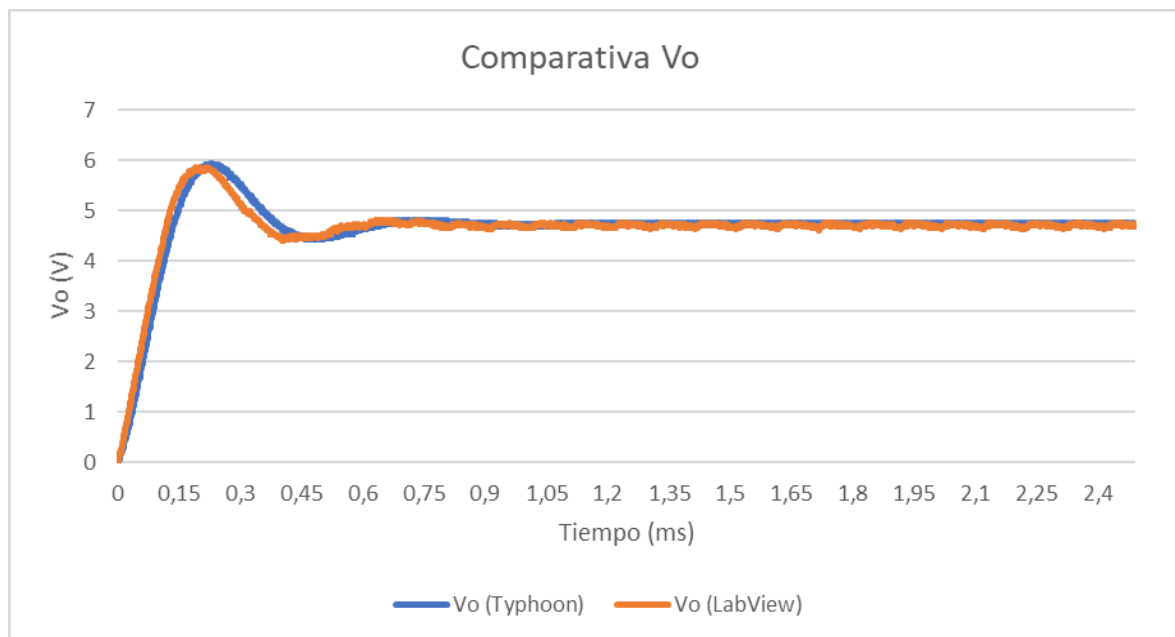


Figura 4-59: Comparativa V_o Typhoon vs LabVIEW (con pérdidas)

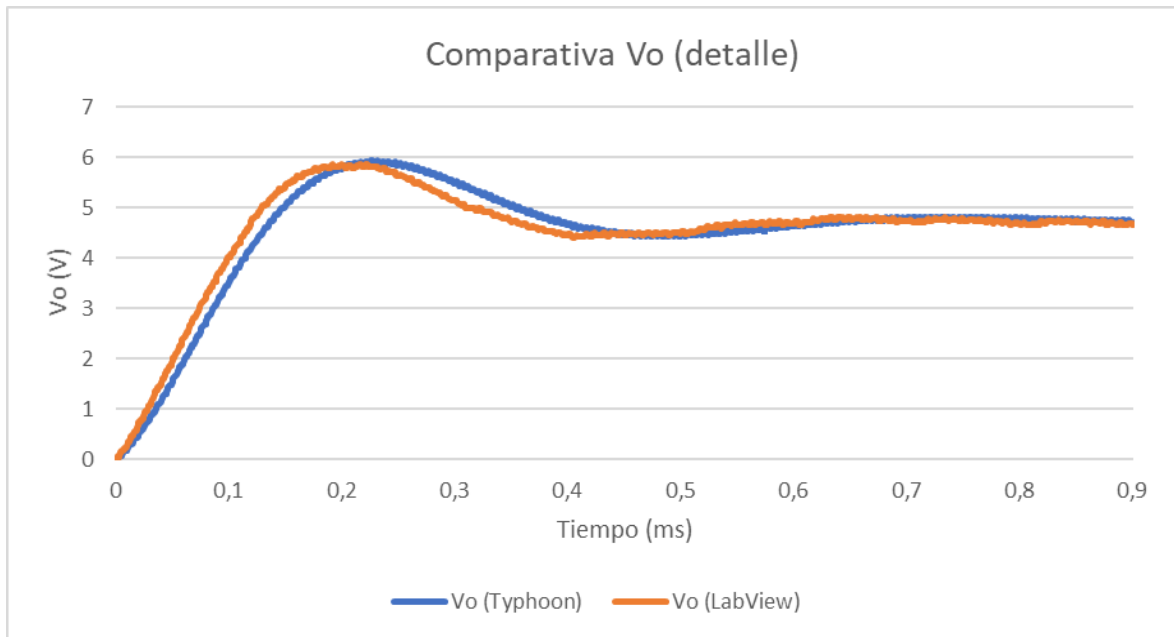


Figura 4-60: Detalle comparativa V_o Typhoon vs LabVIEW (con pérdidas)

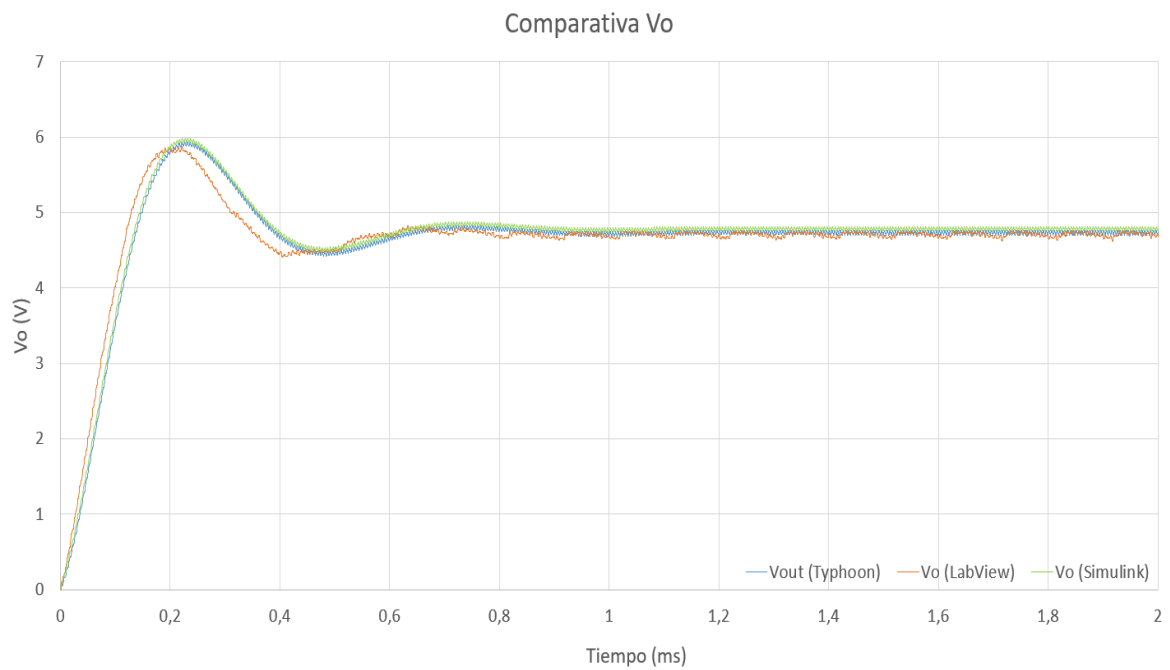


Figura 4-61: Comparativa V_o Typhoon vs LabVIEW vs Simulink (con pérdidas)

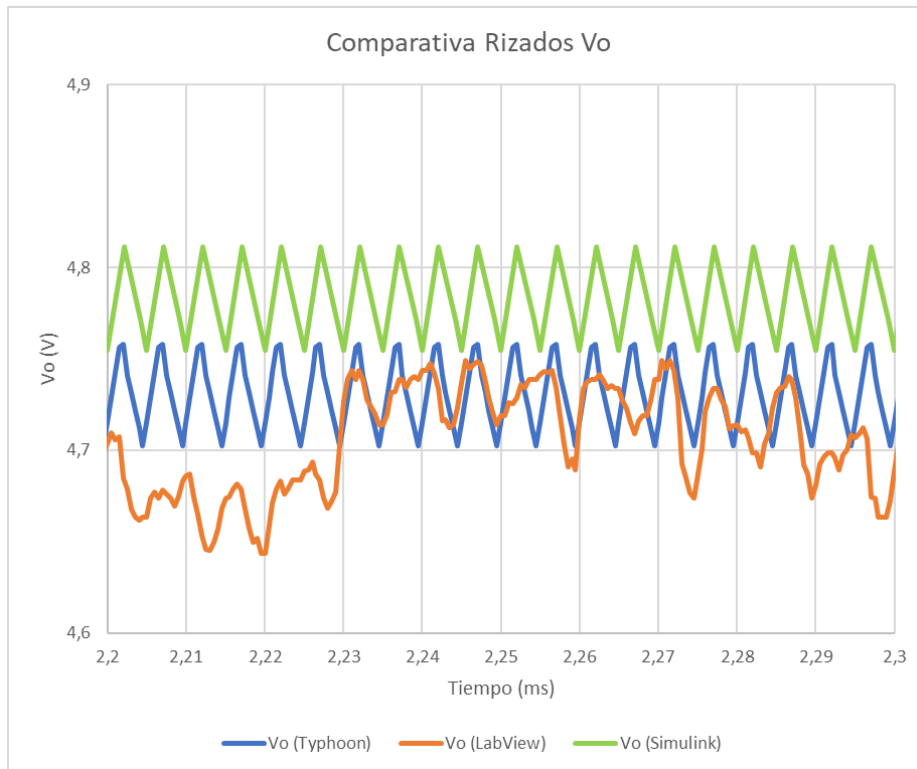


Figura 4-62: Comparativa rizados V_o Typhoon vs LabVIEW vs Simulink (con pérdidas)

Las pérdidas hacen que la forma de la señal varíe, pareciendo apenas un segundo grado, y que los niveles de la señal de tensión caigan hasta los 4,78 V (en el modelo teórico) para el régimen permanente. En el transitorio además la sobresocilación máxima pasa de 7,2 V a apenas 6 V. Typhoon HIL sigue comportándose de forma cuasi-ideal, con la salvedad de que es unos 50 mV menor en amplitud. LabVIEW a pesar de presentar ese efecto de rizado ya comentado, algo menos pronunciado en este caso, alcanza valores muy parecidos a las demás soluciones tanto de voltaje como temporales.

- Corriente por la bobina (I_L)

En las Figuras 4-63, 4-64 y 4-65 se comparan las corrientes por la bobina en el escenario con pérdidas. En la Figura 4-66 aparece la comparativa de rizados para una porción del régimen permanente.

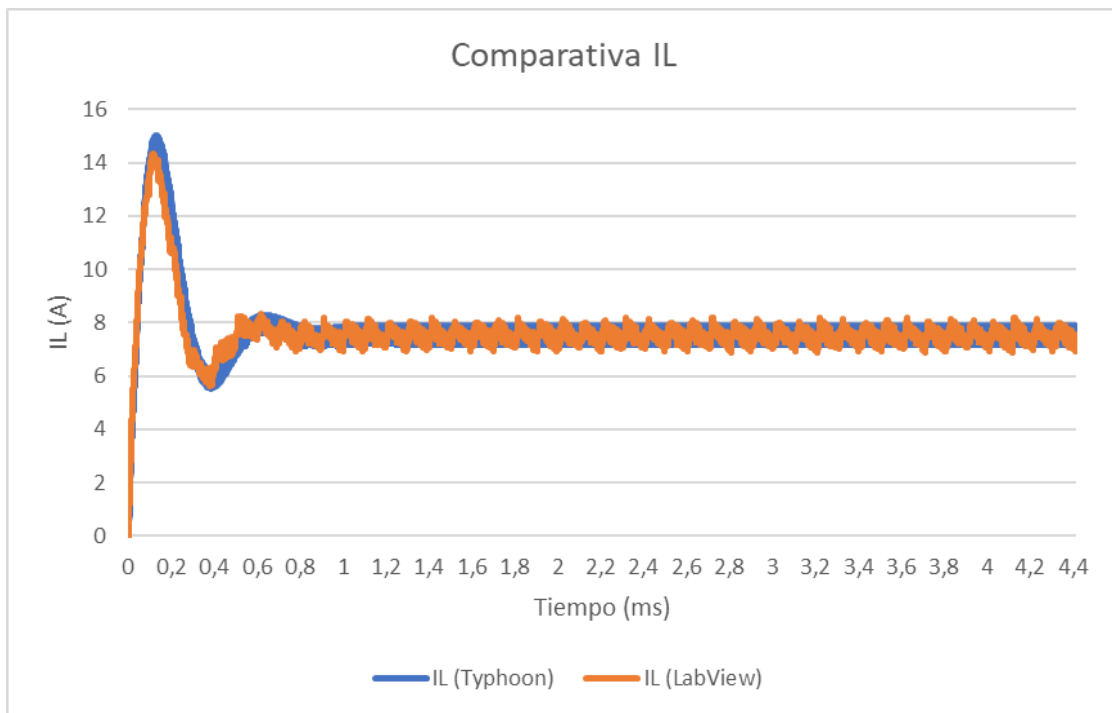


Figura 4-63: Comparativa I_L Typhoon vs LabVIEW (con pérdidas)

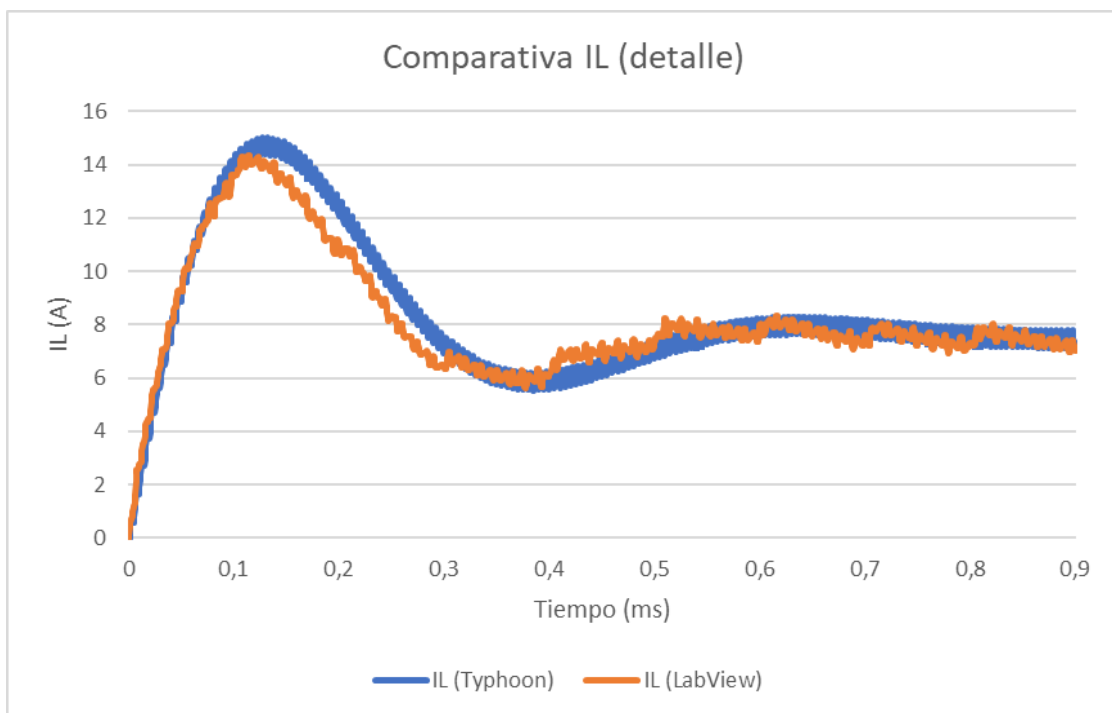


Figura 4-64: Detalle comparativa I_L Typhoon vs LabVIEW (con pérdidas)

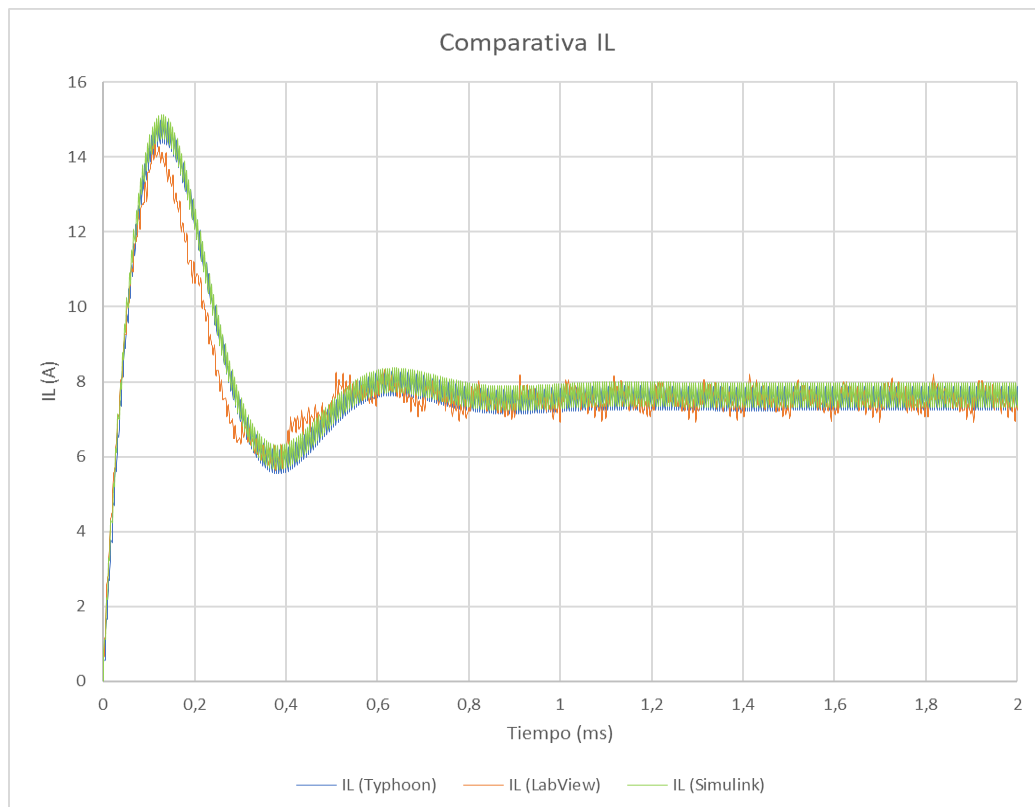


Figura 4-65: Comparativa I_L Typhoon vs LabVIEW vs Simulink (con pérdidas)

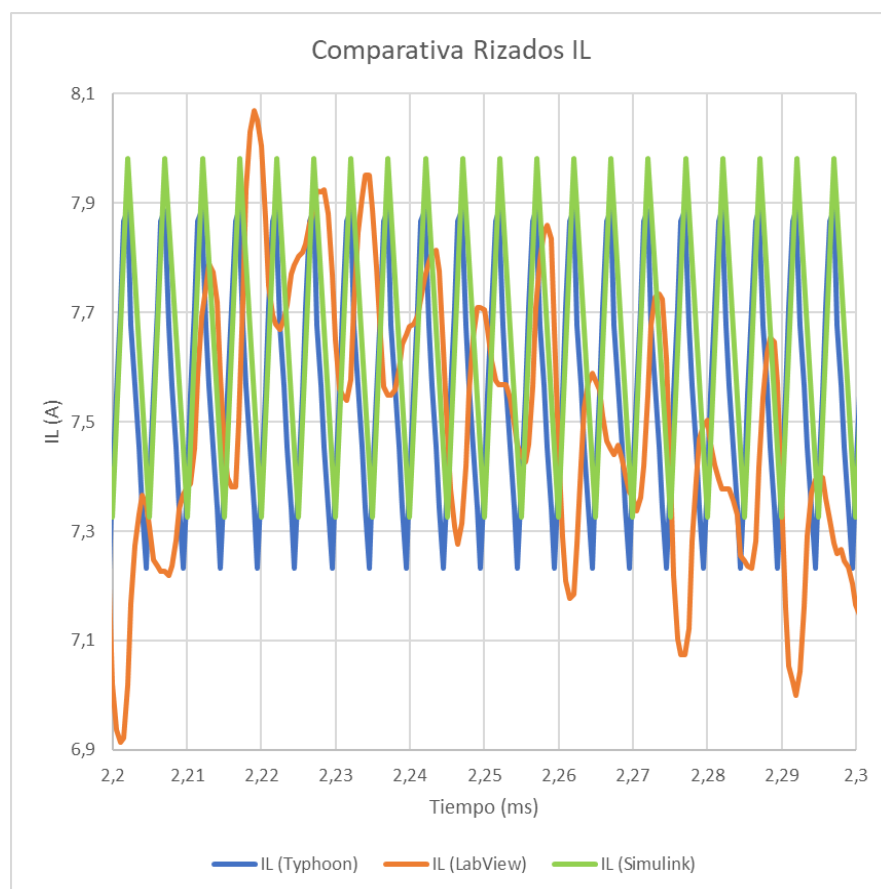


Figura 4-66: Comparativa rizados I_L Typhoon vs LabVIEW vs Simulink (con pérdidas)

Como cabría esperar, la corriente por la bobina también es menor en las tres herramientas que la del escenario sin pérdidas. La sobreoscilación máxima pasa de ser de unos 18 A a ser de apenas unos 15 A. Las dos plataformas comparadas sobre el caso ideal (Typhoon y LabVIEW) se comportan de forma bastante fiel al caso teórico, siendo Typhoon muy preciso y LabVIEW, algo menos por el rizado obtenido.

Además de gráficamente, se ha calculado el error muestra a muestra de los resultados de voltaje y corriente tanto de Typhoon como de LabVIEW, respecto al modelo ideal de Simulink y, posteriormente se ha calculado el sumatorio de dichos errores y se ha dividido entre el número de muestras, para calcular el error medio. Dicha medida aparece en la Tabla 4-8.

		Error V_o	Error I_L
SIN PÉRDIDAS	Typhoon HIL	0,05733 V	0,14671 A
	LabVIEW FPGA	0,10581 V	0,57888 A
CON PÉRDIDAS	Typhoon HIL	0,05344 V	0,14247 A
	LabVIEW FPGA	0,09467 V	0,32531 A

Tabla 4-8: Comparativa error medio V_o e I_L

Del mismo modo, se han calculado los errores medios de la tensión de salida y de la corriente por la bobina del rizado en el régimen permanente. Dichos resultados se muestran en la Tabla 4-9.

		Error V_o	Error I_L
SIN PÉRDIDAS	Typhoon HIL	0,05601 V	0,13987 A
	LabVIEW FPGA	0,07661 V	0,59011 A
CON PÉRDIDAS	Typhoon HIL	0,05211 V	0,13765 A
	LabVIEW FPGA	0,07517 V	0,28569 A

Tabla 4-9: Comparativa error medio de rizados V_o e I_L

Como se puede observar, el error medio de la señal completa no difiere demasiado del cometido al tener en cuenta únicamente el régimen permanente, aunque, en algunos casos, es ligeramente inferior en este segundo caso.

Por último, se muestra en las Tabla 4-10 y 4-11, con mayor detalle, el valor máximo, mínimo, el valor pico a pico y el valor medio del rizado en régimen permanente para cada uno de los casos, tanto de voltaje de salida como de corriente por la bobina, de manera que se pueda visualizar una comparativa de manera no meramente gráfica.

		Valor mínimo	Valor máximo	Valor pico a pico	Valor medio
SIN PÉRDIDAS	Simulink	4,99829 V	5,00079 V	2,53 mV	4,99954 V
	Typhoon HIL	4,94146 V	4,94525 V	3,79 mV	4,94336 V
	LabVIEW FPGA	4,89557 V	4,95462 V	59,05 mV	4,92509 V
CON PÉRDIDAS	Simulink	4,75471 V	4,81116 V	56,54 mV	4,78293 V
	Typhoon HIL	4,70209 V	4,75783 V	55,74 mV	4,72996 V
	LabVIEW FPGA	4,64337 V	4,75863 V	115,26 mV	4,70100 V

Tabla 4-10: Comparativa de rizados V_o

		Valor mínimo	Valor máximo	Valor pico a pico	Valor medio
SIN PÉRDIDAS	Simulink	7,66299 A	8,33135 A	0,66836 A	7,99717 A
	Typhoon HIL	7,56426 A	8,23093 A	0,66667 A	7,89759 A
	LabVIEW FPGA	6,78470 A	7,95933 A	1,17463 A	7,37202 A
CON PÉRDIDAS	Simulink	7,32599 A	7,98049 A	0,65450 A	7,65324 A
	Typhoon HIL	7,23298 A	7,88639 A	0,65341 A	7,55969 A
	LabVIEW FPGA	6,91488 A	8,06878 A	1,15390 A	7,49183 A

Tabla 4-11: Comparativa de rizados I_L

A la vista de los resultados recogidos en las tablas anteriores, se puede inferir nuevamente que los resultados en la plataforma de Typhoon HIL alcanzan valores más precisos tanto en valores mínimo y máximo como en valor medio, de voltaje y corriente, respecto al caso ideal en Simulink. Además, el valor pico a pico es prácticamente igual para Simulink y Typhoon, mientras que es de casi el doble para LabVIEW. Esto es así tanto para el escenario sin pérdidas como para el escenario con pérdidas.

4.4.3 LabVIEW simulaciones a mínimo “dt”

Tal y como se ha comentado en apartados anteriores, Typhoon HIL ha sido limitante a la hora de escoger el paso de simulación, ya que permite como mínimo 500 ns. La solución diseñada en LabVIEW FPGA permite como se ha visto en el apartado 4.2.4 un paso mínimo de 150 ns y se ha tenido que amoldar a los tiempos de Typhoon.

A continuación, se muestran los resultados tanto de tensión como de corriente obtenidos en dicha plataforma para su mínimo “dt”. Mientras que la Figura 4-67 muestra los resultados de V_O con y sin pérdidas y la Figura 4-68 muestra los rizados en una porción del régimen permanente, las Figuras 4-69 y 4-70 enseñan los resultados equivalentes para el caso de la corriente por la bobina (I_L).

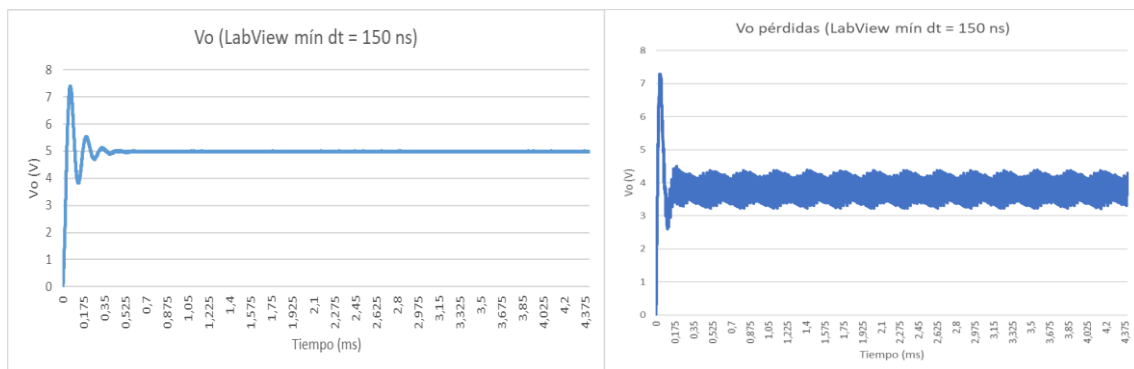


Figura 4-67: V_O LabVIEW mínimo “dt” sin y con pérdidas (izda. y dcha. respectivamente)

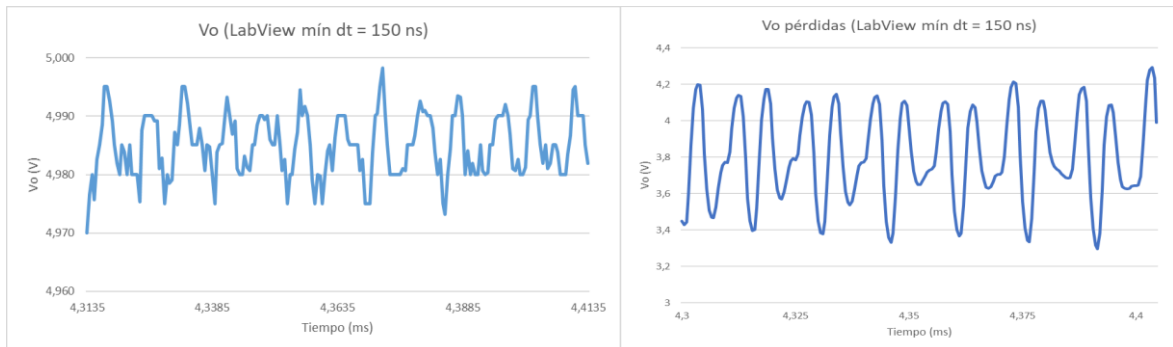


Figura 4-68: Rizado V_O LabVIEW mínimo “dt” sin y con pérdidas (izda. y dcha. respectivamente)

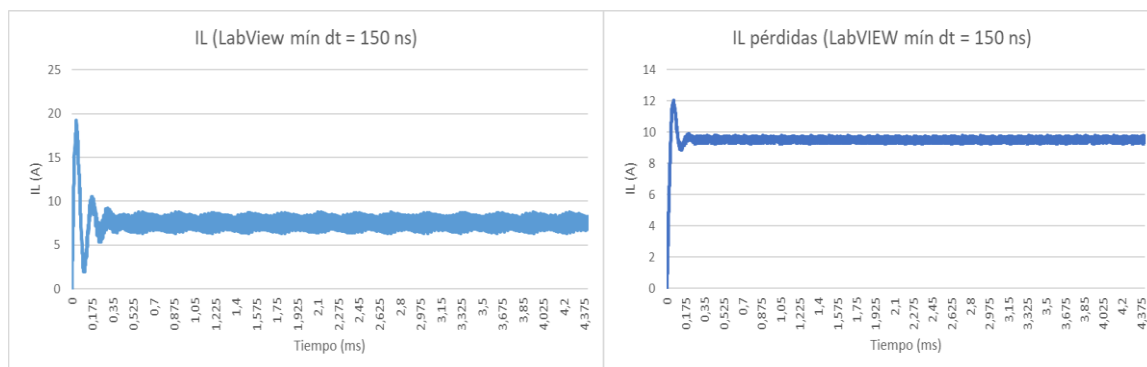


Figura 4-69: I_L LabVIEW mínimo “dt” sin y con pérdidas (izda. y dcha. respectivamente)

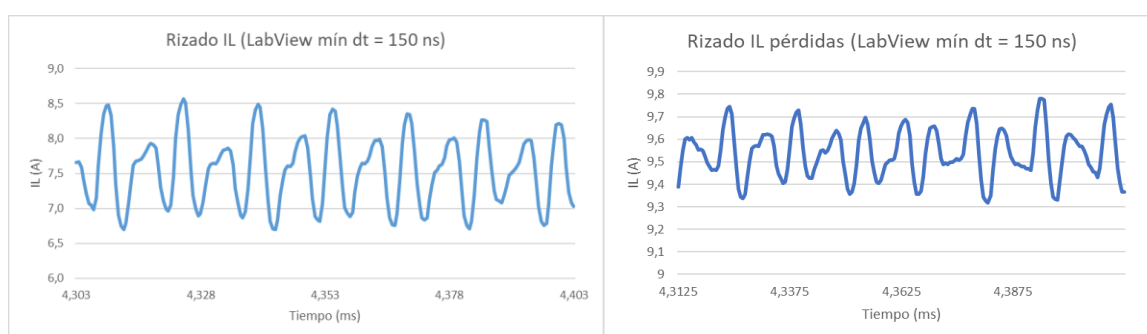


Figura 4-70: Rizado I_L LabVIEW mínimo “dt” sin y con pérdidas (izda. y dcha. respectivamente)

Se puede comprobar como con este “dt” más pequeño, el establecimiento es considerablemente menor y el rizado es algo menos pronunciado. Sin embargo, los valores máximos de sobreoscilación no varían y son idénticos a los del caso de un paso de 500 ns.

Sin embargo, tal y como se recoge en la Tabla 4-12, los valores de tensión de salida y de corriente en régimen permanente sí varían al emplear un “dt” diferente. Esto es especialmente pronunciado en el caso de incluir pérdidas, donde los resultados de voltaje medio de salida llegan a variar casi 1 V y los de corriente, prácticamente 2 A.

			Valor mínimo	Valor máximo	Valor pico a pico	Valor medio
SIN PÉRDIDAS	dt = 150 ns	V _O	4,97000 V	4,99984 V	29,84 mV	4,98492 V
		I _L	6,34657 A	8,74607 A	2,39950 A	7,54632 A
	dt = 500 ns	V _O	4,89557 V	4,95462 V	59,05 mV	4,92509 V
		I _L	6,78470 A	7,95933 A	1,17463 A	7,37202 A
CON PÉRDIDAS	dt = 150 ns	V _O	3,21623 V	4,38709 V	1,17086 V	3,80166 V
		I _L	9,29679 A	9,78423 A	0,48744 A	9,54051 A
	dt = 500 ns	V _O	4,64337 V	4,75863 V	115,26 mV	4,70100 V
		I _L	6,91488 A	8,06878 A	1,15390 A	7,49183 A

Tabla 4-12: Comparativa de rizados V_O e I_L a diferentes “dt” en LabVIEW FPGA

5 Conclusiones y trabajo futuro

5.1 Conclusiones

En este Trabajo de Fin de Máster se ha realizado un estudio de las herramientas comerciales capaces de llevar a cabo simulaciones *Hardware-In-the-Loop* (HIL) ofertadas por las distintas empresas del sector.

Para ello, primero se ha comenzado definiendo la terminología empleada en este campo, correspondiente a los diferentes tipos de simulación: *Real Time*, *on-line/off-line* y *Hardware-In-the-Loop*, y se han mostrado las principales compañías que trabajan con este tipo de tecnología.

Una vez hecho esto, se ha pasado a analizar las herramientas ofertadas por cada empresa para llevar a cabo simulaciones HIL, con el objetivo de encontrar aquellas que cumplan como requisitos el hecho de que el *software* sea gratuito o una versión de prueba y que no sea necesaria la adquisición de una máquina HIL de alto coste. Tras este estudio, se determinó que las dos plataformas principales a comparar serían: Typhoon HIL y LabVIEW FPGA de National Instruments, ya que cumplían todas las condiciones mencionadas. Simulink de MATLAB se descartó, dado que, por un lado su versión Real Time Desktop simulaba sobre el procesador del propio equipo sobre el que se ejecute, con lo que no mostraba resultados de utilización/rendimiento en un *hardware* específico y, por otro lado, Simulink Real Time requiere de un dispositivo HW físico para poder simular, del que no se disponía.

A continuación, se ha llevado a cabo un análisis de las características de ambas herramientas de manera teórica, mediante la observación de las especificaciones y limitaciones, tanto de *software* como de capacidad *hardware*, de cada una de ellas y la familiarización y empleo de las mismas. En este primer estudio se ha determinado que, si bien es cierto que, en términos de potencia y capacidad, Typhoon HIL es bastante superior a LabVIEW, básicamente por el *hardware* sobre el que se ha hecho la comparativa, y además permite un modelado, despliegue y simulación más sencillo e intuitivo, LabVIEW ofrece una profundidad mucho mayor en cuanto a personalización del modelo diseñado, al estar hecho a nivel de puertas lógicas.

Algunos de los parámetros y características que se compararon fueron: los tipos de *solver* disponibles para realizar la simulación, el mínimo paso de simulación o “dt” permitido en cada plataforma (que es de 500 ns en Typhoon HIL y 150 ns en el modelo propuesto de LabVIEW), el modo de adquisición de datos en cada caso (con una exportación más sencilla en Typhoon dado que el dispositivo HIL402 dispone de osciloscopios integrados), las frecuencias máximas de reloj (50 MHz para HIL402 y 40 MHz en myRIO-1900) y de conmutación (200 kHz en HIL402 y configurable en LabVIEW) posibles en cada plataforma, el tipo de simulación (RT, *on-line/off-line*) de los *softwares*, el método para incorporar pérdidas al sistema diseñado (inmediato en Typhoon a través de elementos circuitales, pero más limitado que en LabVIEW, donde las pérdidas han de incluirse en las fórmulas matemáticas del modelo), el formato de los reportes de utilización (Typhoon da reportes con elementos de alto nivel y LabVIEW de bajo nivel) y área ocupada, si la licencia es gratuita o no y la usabilidad de cada plataforma.

Con el objetivo de reforzar esta comparativa, se prosiguió proponiendo un circuito concreto para observar las particularidades de cada herramienta sobre un caso práctico determinado. El circuito diseñado se trata de un convertidor conmutado reductor o *buck*, en el cual, la tensión de salida es inferior a la de entrada, con pérdidas y sin ellas. Dicho convertidor, equivalente para todas las plataformas, se diseñó y simuló mediante una simulación HIL, obteniendo para ambas plataformas los resultados de tensión en la carga y corriente por la bobina.

Una vez vistos estos resultados y el proceso de modelado en sí, se reafirmó la mayor complejidad de LabVIEW FPGA para el diseño, a través de fórmulas matemáticas y bucles, frente al sencillo modelado tipo esquemático de Typhoon HIL. Esta característica hace que un circuito relativamente sencillo sea casi inmediato en Typhoon HIL, pero requiera un proceso de formulación no trivial para LabVIEW, pero, como contrapartida, de este modo se puede personalizar mucho más el modelo en el caso de LabVIEW, como la posibilidad de un mayor número de parámetros de pérdidas frente a los bloques ya predefinidos que ofrece Typhoon, ya que se puede seleccionar qué sucede a cada ciclo, lo que permite ajustar la lógica y los tiempos de forma mucho más detallada. Typhoon HIL ofrece la posibilidad de simular mediante una máquina virtual, lo que facilita en gran medida el proceso, ya que no es necesario adquirir físicamente ningún *hardware* para desplegar simulaciones HIL, mientras que, para LabVIEW, es necesario disponer siempre de una máquina física. En cuanto a documentación, ambas plataformas cuentan con gran cantidad de información al alcance del usuario, pero sí que es cierto que Typhoon HIL dispone de su plataforma de formación gratuita HIL Academy, que es de gran ayuda y además está mucho más centrada y acondicionada a temas de electrónica de potencia, que la información de LabVIEW, que es algo más genérica. Además, Typhoon puede decirse que es más accesible, ya que ofrece una licencia de empleo de su *software* gratuita y la posibilidad de usar la máquina virtual durante un año, mientras que en LabVIEW existen más restricciones en cuanto a licencias y periodos de prueba.

Por último, cabe destacar que los resultados, tanto de tensión como de corriente, obtenidos para el circuito *buck* son muy similares en ambas plataformas al caso teórico. Los resultados de tensión y corriente para Typhoon cuentan con una fidelidad del 98,5% respecto al caso ideal, tanto en el escenario con pérdidas como sin pérdidas; mientras que LabVIEW, si bien es cierto que alcanza un 98% de similitud respecto al caso teórico en términos del voltaje de salida, en cuanto a la corriente, dicho parecido baja hasta el 92,7%. Por ello, los resultados de Typhoon se desvían menos de dicha idealidad y cuentan con menos rizado que los de LabVIEW, esto puede ser debido al hecho de que la máquina HIL402 sobre la que se ha realizado el despliegue de Typhoon cuenta con osciloscopios integrados de los cuales se han exportado las medidas directamente, mientras que el myRIO-1900 sobre el que se ha bajado el modelo de LabVIEW, además de tratarse de un dispositivo de propósito más general, no dispone de los mismos y se ha tenido que conectar a un osciloscopio externo, con sus propio DAC, expuesto a los efectos de las pérdidas y el ruido. Pese a estas diferencias, las dos plataformas cuentan con un buen rendimiento y los resultados se asemejan a los esperados, por ello a la hora de elegir una herramienta u otra no existe una solución cerrada y única para todos los casos y que funcione mejor en todos los escenarios posibles, depende de las necesidades del usuario para escoger la que más se amolde a las mismas.

5.2 Trabajo futuro

Tras el estudio visto a lo largo de todo este Trabajo de Fin de Máster, queda patente que la variedad de herramientas comerciales capaces de realizar simulaciones HIL, actualmente disponibles en el mercado, es suficientemente grande como para intentar abarcar todas ellas en un único proyecto. Es por ello que en este trabajo se ha particularizado el estudio a un subconjunto abaricable de este tipo plataformas, imponiendo una serie de condiciones iniciales de selección, y, por lo tanto, una línea futura muy interesante que queda abierta sería el estudio profundo y el despliegue de una simulación HIL de naturaleza similar, sobre plataformas de pago de otros fabricantes como: Opal-RT, dSpace, RTDS o Speedgoat.

Además, sería interesante tratar de buscar un diseño para el modelo del convertidor conmutado, más simplificado, eficiente y, en definitiva, óptimo, que el propuesto para LabVIEW FPGA, de modo que requiriera una menor área y tuviera una menor utilización de componentes, para comprobar si es posible desplegar varios circuitos *bucks* en una simulación HIL sobre esta plataforma de manera simultánea. También en LabVIEW FPGA se podría intentar realizar la simulación sobre un *hardware* de National Instruments más potente y orientado a HIL que myRIO-1900.

Por último y a la vista de los buenos resultados observados tanto en usabilidad como los obtenidos en la simulación HIL de Typhoon HIL, tendría sentido seguir profundizando en esta herramienta, por ejemplo, comparando los resultados de este trabajo con los que se conseguirían sobre la nueva máquina HIL404 o similares.

Referencias

- [1] Pablo Amor Peinado, “Control en lazo cerrado mediante FPGA con ADC integrado”, Trabajo Fin de Grado, Universidad Autónoma de Madrid (UAM), Escuela Politécnica Superior (EPS), 2018.
- [2] Luis Ibarra, Antonio Rosales, Pedro Ponce, Arturo Molina y Raja Ayyanar, “Overview of Real-Time Simulation as a Supporting Effort to Smart-Grid Attainment”, *Energies MDPI*, vol. 10, no. 6, pp. 1-24, junio 2017.
- [3] Steve Bush, “Mazda home-brews industry first hardware-in-loop car system verification rig”, artículo de *Electronics Weekly*, 2018, fecha de acceso 02/05/2020.
<https://www.electronicsweekly.com/news/design/mazda-home-brews-industry-first-hardware-loop-car-system-verification-rig-2018-08/>
- [4] Rupali Dagade, “Global Hardware in the Loop Market”, *Market Research*, Medium 2020, fecha de acceso 03/05/2020.
<https://medium.com/@rdagade77/global-hardware-in-the-loop-market-d860f75df293>
- [5] Robert Tuffcon, “Hardware-in-the-loop Market Will Accelerate Rapidly with Excellent CAGR in Future with Top Prominent Players like Typhoon, Airbus Group SE, Robert Bosch, National Instruments Corp, Speedgoat GmbH, Honda Aircraft Company”, *The Canton Independent Sentinel*, 2020, fecha de acceso 03/05/2020.
<https://thecantonsentinel.com/news/50655/hardware-in-the-loop-market-will-accelerate-rapidly-with-excellent-cagr-in-future-with-top-prominent-players-like-typhoon-airbus-group-se-robert-bosch-national-instruments-corp-speedgoat-gmbh-hon/>
- [6] Leonel Estrada, Nimrod Vázquez, Joaquín Vaquero, Ángel de Castro y Jaime Arau, “Real-Time Hardware in the Loop Simulation Methodology for Power Converters Using LabVIEW FPGA”, *Energies MDPI*, vol. 13, no. 2, pp. 1-17, enero 2020.
- [7] Typhoon HIL, Typhoon HIL Website, fecha de acceso 18/03/2020.
<https://www.typhoon-hil.com/>
- [8] Typhoon HIL, Typhoon HIL Contact, fecha de acceso 18/03/2020.
<https://www.typhoon-hil.com/contact/>
- [9] Typhoon HIL, Typhoon HIL Academy, fecha de acceso 18/03/2020.
<https://hil.academy/>
- [10] Typhoon HIL, HIL402 vs HIL404, fecha de acceso 18/03/2020.
<https://www.typhoon-hil.com/products/hil404-announcement-page/>
- [11] Typhoon HIL, “Typhoon HIL402 Brochure”, Manual del Sistema HIL402.
- [12] National Instruments, NI Website, fecha de acceso 20/03/2020. <https://www.ni.com/>
- [13] National Instruments, “MyRIO1900 User Guide and Specifications”, manual usuario de MyRIO1900.

- [14] Ali Express, NImyRIO-1900, fecha de acceso 20/03/2020.
<https://es.aliexpress.com/item/32998966006.html>
- [15] National Instruments, Multisim, fecha de acceso 20/03/2020.
<https://www.ni.com/es-es/support/downloads/software-products/download.multisim.html#312060>
- [16] National Instruments, LabVIEW FPGA Module, fecha de acceso 20/03/2020.
<https://www.ni.com/es-es/shop/select/labview-fpga-module>
- [17] Mathworks, Mathworks Website, fecha de acceso 19/05/2020.
<https://es.mathworks.com/>
- [18] MathWorks Help, “Speedgoat Target Computers and I/O Hardware”, fecha de acceso 19/05/2020. <https://es.mathworks.com/help/xpc/supported-hardware.html>
- [19] MathWorks, “Get Started with Simulink Desktop Real-Time”, fecha de acceso 19/05/2020.
<https://es.mathworks.com/help/sldrt/getting-started-with-real-time-windows-target.html>
- [20] MathWorks, “Hardware Support from Simulink Desktop Real-Time”, fecha de acceso 19/05/2020.
<https://es.mathworks.com/hardware-support/simulink-desktop-real-time.html>
- [21] María Murillo Moya, “Control digital de convertidores conmutados mediante LabVIEW”, Trabajo Fin de Máster, Universidad Autónoma de Madrid (UAM), Escuela Politécnica Superior (EPS), 2017.
- [22] National Instruments, “An introduction to High-Throughput DSP in LabVIEW FPGA”, 2020, fecha de acceso 31/05/2020.
<https://www.ni.com/es-es/support/documentation/supplemental/11/an-introduction-to-high-throughput-dsp-in-labview-fpga.html>
- [23] National Instruments, “LabVIEW FPGA Compilation Process: From Run Button to Bitfile”, 2020, fecha de acceso 31/05/2020. <http://www.ni.com/tutorial/9381/en/>

Glosario

DAC	Digital-to-Analog Converter
DSP	Digital Signal Processor
EDO	Ecuación Diferencial Ordinaria
EPS	Escuela Politécnica Superior
ESR	Equivalent Series Resistance
FPGA	Field-Programmable Gate Array
HCTLab	Hardware & Control Technology Laboratory
HIL	Hardware-In-the-Loop
IDE	Integrated Development Environment
IO	Input-Output
LED	Light-Emitting Diode
LUT	Look Up Table
PWM	Pulse-Width Modulation
RAM	Random Access Memory
RDS	Resistance between Drain & Source
RT	Real Time
RTDS	Real Time Digital Simulator
SCADA	Supervisory Control And Data Acquisition
SPICE	Simulation Program with Integrated Circuits Emphasis
TFG	Trabajo Fin de Grado
TFM	Trabajo Fin de Máster
UAM	Universidad Autónoma de Madrid
USB	Universal Serial Bus
VI	Virtual Instrument